

CarSharing Queries

2024-11-09

```
library(getPass)
library(DBI)
library(RPostgres)
con <- DBI::dbConnect(RPostgres::Postgres(), user = "cashu", password = getPass::getPass("Database password"), dbname = "cashdb")
```

```
## Please enter password in TK window (Alt+Tab)
```

Query Solutions

Notes:

- This file has been produced with RMarkdown.
- For some exercises multiple alternative solutions are provided.
- For others, the restrictions have been changed compared to the exercise in order to get more illustrative results.

1

- alle Mitglieder
- all members
- Difficulty: Very easy (+)
- Hint:

```
select * from Member;
```

Displaying records 1 - 10

id	type	membersnr	password	homelocation
1031	CoopMember	1023	just4now	1000
1032	CoopMember	1024	just4now	1002
1033	CoopMember	1025	just4now	1004
1034	CoopMember	1026	just4now	1005
1035	CoopMember	1027	just4now	1006
1036	CoopMember	1028	just4now	1008
1026	CompanyMember	1018	just4now	1000
1027	CompanyMember	1019	just4now	1005
1028	CompanyMember	1020	just4now	1006
1029	CompanyMember	1021	just4now	1007

2

- alle Fahrzeuge
- all vehicles
- Difficulty: Very easy (+)
- Hint:

```
select * from vehicle;
```

Displaying records 1 - 10

id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home	avg:
1004	CompactCar	ZH-1004	Smart	Pure	42323	15279	TRUE	2006-02-01	2029-06-06	796.96	(1001,600.00)	1006	€
1005	CompactCar	ZH-1005	Smart	Pure	3495345	18059	TRUE	2005-11-01	2029-06-06	942.52	(1001,600.00)	1003	€
1014	CompactCar	ZH-1015	Smart	Pulse	435432	19369	TRUE	2006-02-01	2029-06-06	953.80	(1001,600.00)	1008	4
1015	CompactCar	ZH-1016	Smart	Pure	5435434	18452	TRUE	2005-11-01	2029-06-06	979.65	(1001,600.00)	1008	€
1037	CompactCar	ZH-8203	Renault	Twingo	4559023	33472	TRUE	2005-07-01	2029-06-06	1785.53	(1001,600.00)	1005	€
1044	CompactCar	ZH-4001	Mini	Cooper	qwertz01	100	TRUE	2016-12-24	NA	5.50	(1001,700.00)	1000	€
1045	CompactCar	ZH-4005	Peugeot	108	qwertz05	100	TRUE	2016-12-31	NA	4.50	(1001,600.00)	NA	4
1046	CompactCar	ZH-4006	Peugeot	108	qwertz06	200	TRUE	2016-12-31	NA	9.00	(1001,600.00)	1004	4
1047	CompactCar	ZH-4007	Peugeot	108	qwertz07	300	TRUE	2016-12-31	NA	13.50	(1001,600.00)	NA	4
1048	CompactCar	ZH-4008	Peugeot	108	qwertz08	400	TRUE	2016-12-31	NA	18.00	(1001,600.00)	1007	4

3

- alle Personenwagen
- all motor cars
- Difficulty: Very easy (+)
- Hint:

```
select *
from motorcar;
```

Displaying records 1 - 10

id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home	avg:
1004	CompactCar	ZH-1004	Smart	Pure	42323	15279	TRUE	2006-02-01	2029-06-06	796.96	(1001,600.00)	1006	€
1005	CompactCar	ZH-1005	Smart	Pure	3495345	18059	TRUE	2005-11-01	2029-06-06	942.52	(1001,600.00)	1003	€

	id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home	avg:
1014	CompactCar	ZH-1015	Smart	Pulse	435432	19369	TRUE	2006-02-01	2029-06-06	953.80	(1001,600.00)	1008	4	
1015	CompactCar	ZH-1016	Smart	Pure	5435434	18452	TRUE	2005-11-01	2029-06-06	979.65	(1001,600.00)	1008	5	
1037	CompactCar	ZH-8203	Renault	Twingo	4559023	33472	TRUE	2005-07-01	2029-06-06	1785.53	(1001,600.00)	1005	5	
1044	CompactCar	ZH-4001	Mini	Cooper	qwertz01	100	TRUE	2016-12-24	NA	5.50	(1001,700.00)	1000	5	
1045	CompactCar	ZH-4005	Peugeot	108	qwertz05	100	TRUE	2016-12-31	NA	4.50	(1001,600.00)	NA	4	
1046	CompactCar	ZH-4006	Peugeot	108	qwertz06	200	TRUE	2016-12-31	NA	9.00	(1001,600.00)	1004	4	
1047	CompactCar	ZH-4007	Peugeot	108	qwertz07	300	TRUE	2016-12-31	NA	13.50	(1001,600.00)	NA	4	
1048	CompactCar	ZH-4008	Peugeot	108	qwertz08	400	TRUE	2016-12-31	NA	18.00	(1001,600.00)	1007	4	

4

- alle Limousinen
- all limousines
- Difficulty: Very easy (+)
- Hint:

```
select *
from limousine;
```

7 records

	id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home	avg:
1000	Limousine	194 SUA	Cadillac	Eldorado	454994	185333	TRUE	1976-01-01	2022-08-16	54702.32	(1001,1150.00)	1000	1	
1007	Limousine	ZH-1007	Audi	A8	454354	28270	TRUE	2005-10-01	2029-01-11	4002.31	(1001,1150.00)	1006	11	
1008	Limousine	ZH-1008	Audi	A8	534534	14849	TRUE	2006-02-01	2029-07-07	2111.69	(1001,1150.00)	1003	11	
1017	Limousine	ZH-1018	Audi	A8	89e01234	19954	TRUE	2006-01-01	2029-02-28	2929.33	(1001,1150.00)	1008	12	
1028	Limousine	ZH-3462	Lincoln	Towncar	534674	33975	TRUE	2005-03-01	2029-08-06	4812.31	(1001,1150.00)	1004	11	
1030	Limousine	ZH-3465	BMW	325i	534675	13046	TRUE	2006-02-01	2029-09-30	1896.38	(1001,1150.00)	1002	12	
1018	Limousine	ZH-1019	Audi	A8	9054212	30975	TRUE	2005-07-01	2028-12-19	4030.50	(1001,1150.00)	1008	10	

5

- Aktuelles Datum und Uhrzeit
- current date and time
- Difficulty: Easy (++)
- Hint: date and time constants

```
select current_date, current_timestamp, now();
```

1 records

current_date	current_timestamp	now
2024-11-17	2024-11-17 17:00:24	2024-11-17 17:00:24

6

- die vollen Namen aller Personen (Wenn nicht anders spezifiziert, ist ab jetzt mit "Name" immer der volle Name, d.h. "vorname nachname", gemeint)
- the full names of all persons (unless specified otherwise, from now on the full name, "firstname lastname" is required whenever "name" is mentioned)
- Difficulty: Very easy (+)

- Hint: string operators

```
select firstname || ' ' || lastname from Person;
```

Displaying records 1 - 10

?column?

Sundance Kid

Lew Archer

Charly Brown

Susie Diamond

Peppermint Betty

Markus Kradolfer

Wyatt Earp

Luca Geppert

Jonathan Geppert

Freddie Freeloader

7

- die Namen aller persönlichen Mitglieder
- the names of all person members
- Difficulty: Very easy (+)
- Hint: string operators

```
select firstname || ' ' || lastname from personmember;
```

Displaying records 1 - 10

?column?

Wyatt Earp

Luca Geppert

Jonathan Geppert

Freddie Freeloader

Butch Cassidy

Klaus Dittrich

Bjoern Malmstroem

Sharon McCone

Martin Beck

Daniel Roth

8

- die Namen und Heimatstationen der persönlichen Mitglieder, sortiert nach Station
- the names and home location of the person members, in the order of the home locations
- Difficulty: Easy (++)
- Hint: join, order by

```
select firstname || ' ' || lastname, name
from personmember p join
location l on p.homelocation = l.id
order by name;
```

Displaying records 1 - 10

?column?

name

Jonathan Geppert

Greifensee

Freddie Freeloader

Greifensee

Luca Geppert

Greifensee

Daniel Roth

Moenchaltorf

Klaus Dittrich

Schwerzenbach

Henry Gondorff

Uster

Johnny Hooker

Uster

Wolfgang Lehner

Uster

Al Capone

Uster

Wyatt Earp

Zurich

9

- die Namen und Heimatstation der Genossenschaftsmitglieder, sortiert nach Namen der Heimatstation
- names and home locations of the coop members sorted by home location name
- Difficulty: Easy (++)
- Hint: join, order by

```
select firstname || ' ' || lastname, name
from coopmember c join
location l on c.homelocation = l.id
order by name;
```

6 records

?column?	name
Sundance Kid	Greifensee
Lew Archer	Moenchaltorf
Susie Diamond	Schwerzenbach
Peppermint Betty	Uster
Markus Kradolfer	Zurich
Charly Brown	Zurich-Seefeld

10

- die Kennzeichen, Marken und Modelle aller Fahrzeuge
- the licenseplate, make, and model of all vehicles
- Difficulty: Very easy (+)
- Hint:

```
select licensePlate, make, model from vehicle;
```

Displaying records 1 - 10

licenseplate	make	model
ZH-1004	Smart	Pure
ZH-1005	Smart	Pure
ZH-1015	Smart	Pulse
ZH-1016	Smart	Pure
ZH-8203	Renault	Twingo
ZH-4001	Mini	Cooper
ZH-4005	Peugeot	108
ZH-4006	Peugeot	108
ZH-4007	Peugeot	108
ZH-4008	Peugeot	108

11

- die Marken und Modelle der Fahrzeuge mit Namen der Heimatstation
- Make, model, and name of the home location of the vehicles
- Difficulty: Easy (++)
- Hint: join

```
select make, model, name
from vehicle v join
location l on v.home = l.id;
```

Displaying records 1 - 10

make	model	name
Smart	Pure	Uster
Smart	Pure	Zurich-Oerlikon
Smart	Pulse	Zurich
Smart	Pure	Zurich
Renault	Twingo	Schwerzenbach
Mini	Cooper	Greifensee
Peugeot	108	Zurich-Seefeld
Peugeot	108	Zurich-Wipkingen
Chrysler	Le Baron	Moenchaltorf
Ford	Thunderbird	Zurich-Irchel

12

- die Marken und Modelle der Fahrzeuge mit dem Namen der Heimatstation. Auch die Fahrzeuge ohne Heimatstation sollen angezeigt werden.
- Make, model, and name of the home location of the vehicles. The make and model of vehicles without home location should be shown as well
- Difficulty: Moderate (+++)
- Hint: use the adequate join

```
select make, model, name
from vehicle v left outer join
location l on v.home = l.id;
```

Displaying records 1 - 10

make	model	name
Smart	Pure	Uster
Smart	Pure	Zurich-Oerlikon
Smart	Pulse	Zurich
Smart	Pure	Zurich

make	model	name
Renault	Twingo	Schwerzenbach
Mini	Cooper	Greifensee
Peugeot	108	NA
Peugeot	108	Zurich-Seefeld
Peugeot	108	NA
Peugeot	108	Zurich-Wipkingen

13

- die Marken und Modelle der Fahrzeuge mit Heimatstation. Auch die Fahrzeuge ohne Heimatstation und die Stationen ohne Fahrzeuge sollen angezeigt werden.
- Make, model, and name of the home location of the vehicles. The make and model of vehicles without home location as well as the locations without vehicles should be shown as well
- Difficulty: Moderate (+++)
- Hint: use the adequate join

```
select make, model, name
from vehicle v full outer join
Location l on v.home = l.id;
```

Displaying records 1 - 10

make	model	name
Smart	Pure	Uster
Smart	Pure	Zurich-Oerlikon
Smart	Pulse	Zurich
Smart	Pure	Zurich
Renault	Twingo	Schwerzenbach
Mini	Cooper	Greifensee
Peugeot	108	NA
Peugeot	108	Zurich-Seefeld
Peugeot	108	NA
Peugeot	108	Zurich-Wipkingen

14

- Fahrzeuge ohne Heimatstation
- Vehicles without a home location
- Difficulty: Easy (++)
- Hint: Which value would the home location have in such a case? And How do you test this?

```
select *
from vehicle
where home is NULL;
```

3 records

id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home	avgg
1045	CompactCar	ZH-4005	Peugeot	108	qwertz05	100	TRUE	2016-12-31	NA	4.5	(1001,600.00)	NA	.
1047	CompactCar	ZH-4007	Peugeot	108	qwertz07	300	TRUE	2016-12-31	NA	13.5	(1001,600.00)	NA	.
1049	CompactCar	ZH-4009	Peugeot	108	qwertz09	500	TRUE	2016-12-31	NA	22.5	(1001,600.00)	NA	.

15

- Stationen ohne Fahrzeuge
- Locations without vehicles
- Difficulty: Moderate (+++)
- Hint: Take a look at the previous three queries

```
insert into location(id, shortname, name, address)
values(nextval('locationseq'), 'lin', 'Uster-Linde', row('Aathalstrasse', 50, '8610', 'Uster'));
```

```
select shortname, name
from vehicle v right outer join
location l on v.home = l.id
where v.id is null;
```

Displaying records 1 - 10

shortname	name
lin	Uster-Linde
lin	Uster-Linde
lin	Uster-Linde
lin	Uster-Linde
lin	Uster-Linde
lin	Uster-Linde
lin	Uster-Linde
lin	Uster-Linde

shortname	name
lin	Uster-Linde
lin	Uster-Linde

16

- die Kuerzel, Namen und Adressen der Stationen
- short name, long name, and address of the locations
- Difficulty: Very easy (+)
- Hint:

```
select shortName, name, address
from location;
```

Displaying records 1 - 10

shortname	name	address
lei	Zurich-Leimbach	(Grossackerstrasse,32,8041,Zuerich)
oer	Zurich-Oerlikon	(Gubelstrasse,6,8011,Zuerich)
irc	Zurich-Irchel	(Winterthurerstrasse,197,8057,Irchel)
wip	Zurich-Wipkingen	(Weizenstrasse,6,8037,Zuerich)
see	Zurich-Seefeld	(Dufourstrasse,175,8008,Zuerich)
moe	Moenchaltorf	("Im Winkel",4,69123,Moenchaltorf)
zrh	Zurich	(Bahnhofstrasse,101,8000,Zuerich)
ust	Uster	(Centralstrasse,9,8610,Uster)
lin	Uster-Linde	(Aathalstrasse,50,8610,Uster)
lin	Uster-Linde	(Aathalstrasse,50,8610,Uster)

17

- die Kuerzel, Namen und Orte der Stationen
- short name, long name, and city of the locations
- Difficulty: Easy (++)
- Hint: how to access the city attribute?

```
select shortName, name, (address).city
from location;
```

Displaying records 1 - 10

shortname	name	city
lei	Zurich-Leimbach	Zuerich
oer	Zurich-Oerlikon	Zuerich
irc	Zurich-Irchel	Irchel
wip	Zurich-Wipkingen	Zuerich
see	Zurich-Seefeld	Zuerich
moe	Moenchaltorf	Moenchaltorf
zrh	Zurich	Zuerich
ust	Uster	Uster
lin	Uster-Linde	Uster
lin	Uster-Linde	Uster

18

- die Reservationen des Fahrzeugs 'ZH-1020'
- the reservation of vehicle 'ZH-1020'
- Difficulty: Easy (++)
- Hint: simple where clause

```
select v.licenseplate, r.resnumber, r.interval
from vehicle v join
reservation r on r.vehicle = v.id
where v.licensePlate = 'ZH-1020';
```

Displaying records 1 - 10

licenseplate	resnumber	interval
ZH-1020	253	("2024-08-13 12:00:00","2024-08-14 08:00:00")
ZH-1020	254	("2024-09-09 12:00:00","2024-09-09 20:00:00")
ZH-1020	255	("2024-10-03 04:00:00","2024-10-04 04:00:00")
ZH-1020	256	("2024-10-06 12:00:00","2024-10-06 20:00:00")
ZH-1020	436	("2024-09-13 12:00:00","2024-09-14 08:00:00")
ZH-1020	437	("2024-10-09 12:00:00","2024-10-09 20:00:00")
ZH-1020	438	("2024-11-04 04:00:00","2024-11-05 04:00:00")
ZH-1020	439	("2024-11-07 12:00:00","2024-11-07 20:00:00")
ZH-1020	619	("2024-10-13 12:00:00","2024-10-14 08:00:00")

licenseplate	resnumber	interval
ZH-1020	620	("2024-11-10 12:00:00","2024-11-10 20:00:00")

19

- die Reservationen des Mitglieds mit der Nummer 1000
- the reservations of the member with member number 1000
- Difficulty: Easy (++)
- Hint: simple where clause

```
select m.memberNr, r.resnumber, r.interval
from reservation r join
      member m on r.member = m.id
where m.memberNr = 1000;
```

Displaying records 1 - 10

membernr	resnumber	interval
1000	3208	("2024-11-12 10:00:00","2024-11-12 12:00:00")
1000	3203	("2025-06-24 20:00:00","2025-06-24 22:00:00")
1000	1938	("2025-06-07 06:00:00","2025-06-07 11:00:00")
1000	96	("2024-07-29 14:00:00","2024-07-29 23:00:00")
1000	97	("2024-08-20 20:00:00","2024-08-21 02:00:00")
1000	108	("2024-08-01 06:00:00","2024-08-01 11:00:00")
1000	160	("2024-07-24 04:00:00","2024-07-24 14:00:00")
1000	163	("2024-08-03 00:00:00","2024-08-03 08:00:00")
1000	173	("2024-07-20 20:00:00","2024-07-21 16:00:00")
1000	175	("2024-07-25 00:00:00","2024-07-25 11:00:00")

20

- die offenen (zukünftigen) Reservationen des Mitglieds mit der Nummer 1000
- open (future) reservations of the member number 1000
- Difficulty: Easy (++)
- Hint: date/time comparison

```
select m.memberNr, r.resnumber, r.interval
from reservation r join
      member m on r.member = m.id
where m.memberNr = 1000
      and (r.interval).endts > current_timestamp;
```

Displaying records 1 - 10

membernr	resnumber	interval
1000	3203	("2025-06-24 20:00:00","2025-06-24 22:00:00")
1000	1938	("2025-06-07 06:00:00","2025-06-07 11:00:00")
1000	646	("2024-11-21 20:00:00","2024-11-22 02:00:00")
1000	753	("2024-12-11 12:00:00","2024-12-11 20:00:00")
1000	828	("2024-11-28 14:00:00","2024-11-28 23:00:00")
1000	829	("2024-12-21 20:00:00","2024-12-22 02:00:00")
1000	840	("2024-12-02 06:00:00","2024-12-02 11:00:00")
1000	892	("2024-11-26 04:00:00","2024-11-26 14:00:00")
1000	895	("2024-12-04 00:00:00","2024-12-04 08:00:00")
1000	905	("2024-11-22 20:00:00","2024-11-23 16:00:00")

21

- alle Fahrzeuge in Zürich
- all vehicles whose home location is in the city of Zurich
- Difficulty: Easy (++)
- Hint: Once more, access to an attribute of a structured type

```
select v.*
from vehicle v join
      location l on v.home = l.id
where (l.address).city = 'Zuerich';
```

Displaying records 1 - 10

id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home
1005	CompactCar	ZH-1005	Smart	Pure	3495345	18059	TRUE	2005-11-01	2029-06-06	942.52	(1001,600.00)	1003
1014	CompactCar	ZH-1015	Smart	Pulse	435432	19369	TRUE	2006-02-01	2029-06-06	953.80	(1001,600.00)	1008
1015	CompactCar	ZH-1016	Smart	Pure	5435434	18452	TRUE	2005-11-01	2029-06-06	979.65	(1001,600.00)	1008
1046	CompactCar	ZH-4006	Peugeot	108	qwertz06	200	TRUE	2016-12-31	NA	9.00	(1001,600.00)	1004
1048	CompactCar	ZH-4008	Peugeot	108	qwertz08	400	TRUE	2016-12-31	NA	18.00	(1001,600.00)	1007
1009	Convertible	ZH-1009	Alfa Romeo	Spider	349898	17783	TRUE	2005-12-01	2029-07-08	1868.37	(1001,850.00)	1004
1019	Convertible	ZH-1020	Ford	Mustang	9529343	22651	TRUE	2005-06-01	2028-12-20	2382.85	(1001,850.00)	1008

id	type	licenseplate	make	model	motornumber	kilometers	isoperational	operationstart	lastservice	gasconsumption	insurance	home
1043	Convertible	ZH-5432	Chevrolet	Corvette	qwerty99	0	TRUE	2016-10-03	NA	0.00	(1001,2000.00)	1009
1008	Limousine	ZH-1008	Audi	A8	534534	14849	TRUE	2006-02-01	2029-07-07	2111.69	(1001,1150.00)	1003
1017	Limousine	ZH-1018	Audi	A8	89e01234	19954	TRUE	2006-01-01	2029-02-28	2929.33	(1001,1150.00)	1008

22a

- die Anzahl Limousinen pro Station
- the locations and number of limousines at each location
- Difficulty: Easy (++)
- Hint:

```
select shortname, name, count(*)
  from limousine m join
        location l on m.home = l.id
  group by shortname, name
union
select shortname, name, 0
  from limousine m right outer join
        location l on m.home = l.id
 where m.licenseplate is null;
```

Displaying records 1 - 10

shortname	name	count
lei	Zurich-Leimbach	0
zrh	Zurich	2
wip	Zurich-Wipkingen	0
ust	Uster	1
grs	Greifensee	1
swb	Schwerzenbach	0
see	Zurich-Seefeld	1
irc	Zurich-Irchel	0
moe	Moenchaltorf	1
oer	Zurich-Oerlikon	1

22

```
select shortname, name, count(m.id)
  from limousine m right outer join
        location l on m.home = l.id
  group by shortname, name;
```

Displaying records 1 - 10

shortname	name	count
irc	Zurich-Irchel	0
lin	Uster-Linde	0
moe	Moenchaltorf	1
grs	Greifensee	1
ust	Uster	1
wip	Zurich-Wipkingen	0
swb	Schwerzenbach	0
see	Zurich-Seefeld	1
lei	Zurich-Leimbach	0
oer	Zurich-Oerlikon	1

23

- alle offenen (unbezahlten) Rechnungen
- all open invoices (unpaid)
- Difficulty: Easy (++)
- Hint:

```
select * from invoice where not ispaid;
```

Displaying records 1 - 10

id	invoicedate	duedate	invoicenumber	ispaid	member	total
1329	2024-07-01	2024-07-31	1329	FALSE	1009	40.34
1330	2024-07-01	2024-07-31	1330	FALSE	1019	97.11
1331	2024-08-01	2024-08-31	1331	FALSE	1008	903.52
1332	2024-08-01	2024-08-31	1332	FALSE	1009	1612.77
1333	2024-08-01	2024-08-31	1333	FALSE	1011	913.61
1334	2024-08-01	2024-08-31	1334	FALSE	1012	1648.23
1335	2024-08-01	2024-08-31	1335	FALSE	1013	1185.59
1336	2024-08-01	2024-08-31	1336	FALSE	1014	177.37

id	invoicedate	duedate	invoicenumber	ispaid	member	total
1337	2024-08-01	2024-08-31	1337	FALSE	1015	732.75
1338	2024-08-01	2024-08-31	1338	FALSE	1016	340.50

24

- alle ueberfaelligen Rechnungen
- all overdue invoices
- Difficulty: Easy (++)
- Hint: Once more date/time comparisons

```
select *
  From invoice
 where not ispaid
        and dueDate < current_date;
```

Displaying records 1 - 10

id	invoicedate	duedate	invoicenumber	ispaid	member	total
1329	2024-07-01	2024-07-31	1329	FALSE	1009	40.34
1330	2024-07-01	2024-07-31	1330	FALSE	1019	97.11
1331	2024-08-01	2024-08-31	1331	FALSE	1008	903.52
1332	2024-08-01	2024-08-31	1332	FALSE	1009	1612.77
1333	2024-08-01	2024-08-31	1333	FALSE	1011	913.61
1334	2024-08-01	2024-08-31	1334	FALSE	1012	1648.23
1335	2024-08-01	2024-08-31	1335	FALSE	1013	1185.59
1336	2024-08-01	2024-08-31	1336	FALSE	1014	177.37
1337	2024-08-01	2024-08-31	1337	FALSE	1015	732.75
1338	2024-08-01	2024-08-31	1338	FALSE	1016	340.50

25

- alle Reservationen, die noch nicht in Rechnung gestellt wurden
- all reservations that have not yet been billed
- Difficulty: Easy (++)
- Hint: null test

```
select resnumber
  from reservation r join
    Useofvehicle u on u.reservation = r.id
 where invoice is null;
```

Displaying records 1 - 10

resnumber
862
868
193
224
255
256
332
336
339
368

26

- die Rechnungen des Mitglieds mit der Nummer 1000
- the invoices for member number 1000
- Difficulty: Easy (++)
- Hint:

```
select *
  from invoice i join
    member m on i.member = m.id
 where m.memberNr = 1000;
```

3 records

id	invoicedate	duedate	invoicenumber	ispaid	member	total	id	type	membernr	password	homelocation
1331	2024-08-01	2024-08-31	1331	FALSE	1008	903.52	1008	PersonMember	1000	luckyluke	1000
1353	2024-09-01	2024-10-01	1353	FALSE	1008	1141.61	1008	PersonMember	1000	luckyluke	1000
1381	2024-10-01	2024-10-31	1381	FALSE	1008	1776.18	1008	PersonMember	1000	luckyluke	1000

27

- die Kennzeichen der Fahrzeuge der Benutzungen
- the license plates of the vehicles which have been used (not only reserved)
- Difficulty: Easy (++)
- Hint:

```
select usageNumber, licensePlate
  from vehicle v join
    reservation r on r.vehicle = v.id join
    useofvehicle u on u.reservation = r.id
 order by usageNumber;
```

Displaying records 1 - 10

usagenumber	licenseplate
30	ZH-1002
8253	194 SUA
8254	194 SUA
8362	ZH-1003
8393	ZH-1011
8424	ZH-1020
8425	ZH-1020
8501	ZH-8204
8505	ZH-8205
8508	ZH-8206

28

- die Namen der persönlichen Mitglieder zusammen mit der Angabe, ob sie bereits einen Unfall hatten
- the names of the person members plus the indication if the members have had an accident
- Difficulty: Easy (++)
- Hint:

```
select firstname || ' ' || lastname, hadAccident from PersonMember;
```

Displaying records 1 - 10

?column?	hadaccident
Wyatt Earp	FALSE
Luca Geppert	FALSE
Jonathan Geppert	FALSE
Freddie Freeloader	TRUE
Butch Cassidy	FALSE
Klaus Dittrich	FALSE
Bjoern Malmstroem	FALSE
Sharon McCone	FALSE
Martin Beck	TRUE
Daniel Roth	FALSE

29

- die Namen der persönlichen Mitglieder zusammen mit der Angabe, ob sie bereits einen Unfall hatten (schoen formattiert)
- the names of the person members plus the indication if the members have had an accident (nicely formatted)
- Difficulty: Moderate (+++)
- Hint: case statement

```
select firstname || ' ' || lastname as name,
       case hadAccident when true THEN 'Unfall'
                    else 'kein Unfall'
       end as hadAccident
  from PersonMember;
```

Displaying records 1 - 10

name	hadaccident
Wyatt Earp	kein Unfall
Luca Geppert	kein Unfall
Jonathan Geppert	kein Unfall
Freddie Freeloader	Unfall
Butch Cassidy	kein Unfall
Klaus Dittrich	kein Unfall
Bjoern Malmstroem	kein Unfall
Sharon McCone	kein Unfall
Martin Beck	Unfall
Daniel Roth	kein Unfall

30

- welche persönlichen Mitglieder hatten bereits einen Unfall?
- which person members have had an accident?
- Difficulty: Easy (++)
- Hint: boolean attribute in where-clause

```
select firstname || ' ' || lastname as Unfallfahrer
from PersonMember
where hadAccident;
```

Displaying records 1 - 10

unfallfahrer

Freddie Freeloader

Martin Beck

Pig Ben

Billy The Kid

Doc Holiday

Al Capone

Sam Spade

Johnny Hooker

Doyle Lonnegan

Cincinnati Kid

31

- die den einzelnen Benutzungen zugrundeliegenden Kilometerpreise
- the correct price per kilometer for each use
- Difficulty: Moderate (+++)
- Hint: Joins (maybe more than one ?)

```
select usageNumber, licensePlate, ppKilometer
from vehicle v join
  reservation r on r.vehicle = v.id join
  useofvehicle u on u.reservation = r.id join
  vehicleStaticData vsd on v.type = vsd.Type
order by usageNumber;
```

Displaying records 1 - 10

usagenumber	licenseplate	ppkilometer
30	ZH-1002	0.56
8253	194 SUA	0.82
8254	194 SUA	0.82
8362	ZH-1003	0.82
8393	ZH-1011	0.88
8424	ZH-1020	0.82
8425	ZH-1020	0.82
8501	ZH-8204	0.82
8505	ZH-8205	0.66
8508	ZH-8206	0.82

32

- Entfernungsbasierte Kosten der Benutzungen
- the distance-based prices for the vehicle uses
- Difficulty: Moderate (+++)
- Hint: Joins (maybe more than one ?)

```
select usageNumber, u.kilometers * ppkilometer as kilometerCost
from vehicle v join
  reservation r on r.vehicle = v.id join
  useofvehicle u on u.reservation = r.id join
  vehicleStaticData vsd on v.type = vsd.Type
order by usageNumber;
```

Displaying records 1 - 10

usagenumber	kilometercost
30	112.00
8253	95.12
8254	48.38
8362	370.64
8393	457.60
8424	1082.40
8425	216.48
8501	1033.20
8505	646.80
8508	1385.80

33

- Die Summe der Benutzungsdauern pro Tag
- the sum (of hours) of vehicle uses per day

- Difficulty: Moderate (+++)
- Hint: how to extract the day/month/year component from a date?

```
select (r.interval).begins :: date, sum(hours)
from useofvehicle u join reservation r on u.reservation = r.id
group by 1
order by 1;
```

Displaying records 1 - 10

begins	sum
2024-06-26	8
2024-07-04	40
2024-07-07	8
2024-07-08	24
2024-07-12	160
2024-07-13	5
2024-07-14	45
2024-07-16	55
2024-07-17	40
2024-07-18	10

34

- Die durchschnittliche Benutzungsdauer
- The average duration of vehicle use
- Difficulty: Easy (++)
- Hint:

```
select avg(hours)
from UseOfVehicle;
```

1 records

avg
13.1929

35

- die kürzeste Entfernung, die während einer Benutzung zurückgelegt wurde
- The shortest distance (in kilometers) driven in a single vehicle use
- Difficulty: Easy (++)
- Hint:

```
select min(kilometers)
from UseOfVehicle;
```

1 records

min
0

36

- die durchschnittliche Entfernung pro Benutzung
- the average distance driven per vehicle use
- Difficulty: Easy (++)
- Hint:

```
select avg(kilometers)
from UseOfVehicle;
```

1 records

avg
443.44

37

- die Summe der Entfernungen, die pro Fahrzeug zurückgelegt wurde
- the sum of distances driven per vehicle
- Difficulty: Easy (++)
- Hint: In the meantime, joins and group-by should be familiar constructs :-)

```
select licenseplate, sum(u.kilometers)
from UseOfVehicle u join
  Reservation r on u.reservation = r.id join
  Vehicle v on r.vehicle = v.id
group by v.licenseplate;
```

Displaying records 1 - 10

licenseplate	sum
ZH-1020	12039.717
ZH-8002	19857.179

licenseplate	sum
ZH-1018	9334.931
ZH-1006	7570.547
ZH-3465	1480.810
ZH-8006	5537.230
ZH-1001	7773.616
ZH-1011	7749.187
ZH-8205	9462.130
ZH-3463	14395.763

38

- die durchschnittliche Entfernungen, die pro Fahrzeug zurückgelegt wurde (in allen Reservationen)
- the average distance driven per vehicle (in all reservations)
- Difficulty: Moderate (+++)
- Hint: First compute the sum of distances per vehicle, and then average that. You can have subqueries in the from-clause, too!

```
select avg(kilometers)
  from (select licenseplate, sum(u.kilometers) as kilometers
        from UseOfVehicle u join
              Reservation r on u.reservation = r.id join
              Vehicle v on r.vehicle = v.id
        group by v.licenseplate) tmp;
```

1 records

avg
11886.35

39

- die Namen und Jahresgebuehren der persoenlichen Mitglieder
- the names and yearly fees of the person members
- Difficulty: Easy (++)
- Hint: join

```
select firstname || ' ' || lastname as name,
       case hadAccident when false then basicFee * 0.8
         else basicFee
       end as yearlyFee
  from PersonMember m join
       MemberFees f on m.type = f.memberType;
```

Displaying records 1 - 10

name	yearlyfee
Wyatt Earp	80
Luca Geppert	80
Jonathan Geppert	80
Freddie Freeloader	100
Butch Cassidy	80
Klaus Dittrich	80
Bjoern Malmstroem	80
Sharon McCone	80
Martin Beck	100
Daniel Roth	80

40

- die Namen und Jahresgebuehren der Firmenmitglieder
- the names and yearly fees of the company members
- Difficulty: Easy (++)
- Hint: join

```
select companyName as name,
       basicFee as yearlyFee
  from companyMember m join
       memberFees f on m.type = f.memberType;
```

6 records

name	yearlyfee
Hair Force	100
Pizza Rocket	100
Dust Busters Unlimited	100
Plump Plumbers	100
Tilting Tilers	100
Psychedelic Painters	100

41

- die Namen und Jahresgebuehren der Genossenschaftsmitglieder
- the names and yearly fees of the coop members
- Difficulty: Easy (++)
- Hint:

```
select firstname || ' ' || lastname as name,
       0 as yearlyFee
from CoopMember m;
```

6 records

name	yearlyfee
Sundance Kid	0
Lew Archer	0
Charly Brown	0
Susie Diamond	0
Peppermint Betty	0
Markus Kradolfer	0

42

- die Namen und Jahresgebuehren aller Mitglieder
- the names and yearly fees of all members
- Difficulty: Moderate (+++)
- Hint: union

```
select companyName as name,
       basicFee as yearlyFee
from companyMember m join
     memberFees f on m.type = f.memberType
union
select firstname || ' ' || lastname as name,
       case hadAccident when true then basicFee * 0.8
         else basicFee
       end as yearlyFee
from PersonMember m join
     memberFees f on m.type = f.memberType
union
select firstname || ' ' || lastname as name,
       0 as yearlyFee
from CoopMember m;
```

Displaying records 1 - 10

name	yearlyfee
Doc Holiday	80
Butch Cassidy	100
Tilting Tilers	100
Hair Force	100
Pig Ben	80
Johnny Hooker	80
Sundance Kid	0
Luca Geppert	100
Jill McBain	100
Sam Spade	80

43

- alle saeumigen Mitglieder
- the members with due and open invoices
- Difficulty: Easy (++)
- Hint:

```
select memberNr
from invoice i join
     member m on i.member = m.id
where not ispaid
     and dueDate < current_date;
```

Displaying records 1 - 10

membernr
1001
1011
1000
1001
1003
1004
1005
1006
1007

memberr

1008

44

- alle saeumigen persoenlichen Mitglieder
- the person members with due and open invoices
- Difficulty: Moderate (+++)
- Hint: Simply join with the appropriate subtable

```
select memberNr
  from invoice i join
        personmember m on i.member = m.id
 where not ispaid
        and dueDate < current_date;
```

Displaying records 1 - 10

memberr

	memberr
	1001
	1011
	1000
	1001
	1003
	1004
	1005
	1006
	1007
	1008

45

- die Namen der saeumigen persoenlichen Mitglieder
- the full names of person members with due and open invoices
- Difficulty: Moderate (+++)
- Hint: Just in case you forgot how to concatenate strings

```
select firstname || ' ' || lastname as name
  from invoice i join
        personmember m on i.member = m.id
 where not ispaid
        and dueDate < current_date
 order by lastname;
```

Displaying records 1 - 10

name
Martin Beck
Martin Beck
Martin Beck
Pig Ben
Pig Ben
Pig Ben
Al Capone
Al Capone
Al Capone
Al Capone

46

- die Namen der saeumigen persoenlichen Mitglieder (nun jedes Mitglied nur einmal !)
- the full names of person members with due and open invoices (make sure that each name appears only once)
- Difficulty: Moderate (+++)
- Hint:

```
select distinct firstname || ' ' || lastname as name
  from invoice i join
        personmember m on i.member = m.id
 where not ispaid
        and dueDate < current_date
 order by 1;
```

Displaying records 1 - 10

name
Al Capone
Billy The Kid
Bjoern Malmstroem
Butch Cassidy
Daniel Roth

name

Doc Holiday

Jonathan Geppert

Kat Colorado

Klaus Dittrich

Kristian Weiser

47

- alle persönlichen Mitglieder, die in einem Ort wohnen, in dem es eine Station mit Limousinen gibt
- all person members living in a city with a location which has limousines
- Difficulty: Moderate (+++)
- Hint: A subquery, but how?

```
select distinct firstname || ' ' || lastname as name,
    (l.address).city as "limo city",
    (m.address).city as "member city"
from personmember m join
    location l on (l.address).city = (m.address).city
where exists (select *
              from limousine
              where home = l.id);
```

Displaying records 1 - 10

name	limo city	member city
Al Capone	Uster	Uster
Bjoern Malmstroem	Zuerich	Zuerich
Butch Cassidy	Zuerich	Zuerich
Daniel Roth	Moenchaltorf	Moenchaltorf
Freddie Freeloader	Greifensee	Greifensee
Kat Colorado	Zuerich	Zuerich
Martin Beck	Zuerich	Zuerich
Pig Ben	Zuerich	Zuerich
Sam Spade	Zuerich	Zuerich
Sharon McCone	Zuerich	Zuerich

48

- die Reservationen fuer Fahrzeuge beliebigen Typs an der station ust
- all reservations for vehicles of any category at the location ust
- Difficulty: Easy (++)
- Hint:

```
select resnumber, interval, licensePlate
from reservation r join
    vehicle v on r.vehicle = v.id join
    location l on v.home = l.id
where l.shortname = 'ust';
```

Displaying records 1 - 10

resnumber	interval	licenseplate
186	("2024-09-03 12:00:00","2024-09-04 12:00:00")	ZH-1002
187	("2024-09-07 13:00:00","2024-09-07 22:00:00")	ZH-1002
188	("2024-09-09 12:00:00","2024-09-09 20:00:00")	ZH-1002
189	("2024-09-10 12:00:00","2024-09-10 20:00:00")	ZH-1002
190	("2024-09-02 04:00:00","2024-09-02 14:00:00")	ZH-1003
191	("2024-09-08 08:00:00","2024-09-09 08:00:00")	ZH-1003
192	("2024-09-10 12:00:00","2024-09-10 20:00:00")	ZH-1003
193	("2024-10-06 12:00:00","2024-10-06 20:00:00")	ZH-1003
194	("2024-08-28 12:00:00","2024-08-28 18:00:00")	ZH-1004
195	("2024-09-10 12:00:00","2024-09-10 20:00:00")	ZH-1004

49a

- die Reservationen fuer Fahrzeuge beliebigen Typs an der station ust am 11.11.2024 von 9-10 Uhr
- all reservations for vehicles of any category at the location ust on 11.11.2024 at 9-10
- Difficulty: Moderate (+++)
- Hint: Predicates with date/time values

```
select resnumber, interval, licensePlate
from reservation r join
    vehicle v on r.vehicle = v.id join
    location l on v.home = l.id
where l.shortname = 'ust'
and ( ( (r.interval).begints <= '2024-11-11 09:00:00'
        and (r.interval).endts > '2024-11-11 09:00:00')
      or
      ( (r.interval).begints >= '2024-11-11 09:00:00'
        and (r.interval).begints <= '2024-11-11 10:00:00'));
```


2 records

resnumber	interval	licenseplate
585	("2024-11-11 00:00:00","2024-11-11 10:00:00")	ZH-1010
588	("2024-11-11 02:00:00","2024-11-11 12:00:00")	ZH-1011

49

- Hint: Overlap operator for range types

```
select resnumber, interval, licensePlate
  from reservation r join
    vehicle v on r.vehicle = v.id join
    location l on v.home = l.id
 where l.shortname = 'ust'
    and tstrange((r.interval).begints, (r.interval).endts, '[')
        && tstrange('2024-11-11 09:00:00', '2024-11-11 10:00:00', '[');
```

2 records

resnumber	interval	licenseplate
585	("2024-11-11 00:00:00","2024-11-11 10:00:00")	ZH-1010
588	("2024-11-11 02:00:00","2024-11-11 12:00:00")	ZH-1011

50a

- alle Fahrzeuge beliebigen Typs, die am 11.11.2024 um 9:00 Uhr frei waren
- all vehicles which are free on 11.11.2024 at 9:00
- Difficulty: Moderate (+++)
- Hint: Predicates with date/time values

```
select licensePlate
  from vehicle v
 where not exists ( select resnumber
                    from reservation r
                    where r.vehicle = v.id
                      and tstrange((r.interval).begints, (r.interval).endts, '[')
                          && tstrange('2024-11-11 09:00:00', '2024-11-11 10:00:00', '['))
 order by 1;
```

Displaying records 1 - 10

licenseplate
194 SUA
ZH-1002
ZH-1003
ZH-1004
ZH-1005
ZH-1006
ZH-1007
ZH-1008
ZH-1009
ZH-1012

50

- alle Fahrzeuge beliebigen Typs, die am 11.11.2024 um 9:00 Uhr frei waren
- all vehicles which are free on 11.11.2024 at 9:00
- Difficulty: Moderate (+++)
- Hint: Predicates with date/time values

```
select licensePlate
  from vehicle v
 where not exists ( select resnumber
                    from reservation r
                    where r.vehicle = v.id
                      and ( ( (r.interval).begints <= '2024-11-11 09:00:00'
                          and (r.interval).endts > '2024-11-11 09:00:00')
                          or
                          ( (r.interval).begints >= '2024-11-11 09:00:00'
                          and (r.interval).begints <= '2024-11-11 10:00:00'))))
 order by 1;
```

Displaying records 1 - 10

licenseplate
194 SUA
ZH-1002
ZH-1003
ZH-1004
ZH-1005
ZH-1006
ZH-1007

licenseplate

ZH-1008

ZH-1009

ZH-1012

51

- alle Fahrzeuge beliebigen Typs an der Station ust, die am 11.11.2024 um 9:00 Uhr frei waren
- all vehicles at location ust which are free on 11.11.2024 at 9:00
- Difficulty: Moderate (+++)
- Hint: Predicates with date/time values

```
select licensePlate
  from vehicle v join
    location l on v.home = l.id
 where shortName = 'ust' and
    not exists ( select resnumber
                 from reservation r
                 where r.vehicle = v.id
                   and tsrange((r.interval).begints, (r.interval).endts, '[')
                   && tsrange('2024-11-11 09:00:00', '2024-11-11 10:00:00', '['))
```

5 records

licenseplate

ZH-1004

ZH-1007

ZH-1002

ZH-1006

ZH-1003

52

- alle PKWs mit Kindersitz an der Station grs, die am 11.11.2024 um 9:00 Uhr frei waren
- all motor cars with a child seat at location ust which are free on 11.11.2024 at 9:00
- Difficulty: Moderate (+++)
- Hint: Predicates with date/time values

```
select licensePlate
  from motorcar v join
    location l on v.home = l.id
 where shortName = 'grs' and
    childSeat and
    not exists ( select resnumber
                 from reservation r
                 where r.vehicle = v.id
                   and tsrange((r.interval).begints, (r.interval).endts, '[')
                   && tsrange('2024-11-11 09:00:00', '2024-11-11 10:00:00', '['))
```

2 records

licenseplate

ZH-3456

ZH-3458

53

- Kennzeichen mit der Angabe, ob das Fahrzeug am 11.11.2024 von 9-10 Uhr frei war
- all license plates together with the information whether the vehicle was free on 11.11.2024 from 9-10
- Difficulty: Moderate (+++)
- Hint: Predicates with date/time values

```
select licensePlate,
       case when (select count(resnumber)
                  from reservation r
                  where r.vehicle = v.id
                    and tsrange((r.interval).begints, (r.interval).endts, '[')
                    && tsrange('2024-11-11 09:00:00', '2024-11-11 10:00:00', '[')) = 0
              then 'frei'
              else 'reserviert'
        end as freiOderBelegt
  from vehicle v
 order by licensePlate;
```

Displaying records 1 - 10

licenseplate	freioderbelegt
194 SUA	frei
ZH-1001	reserviert
ZH-1002	frei
ZH-1003	frei
ZH-1004	frei
ZH-1005	frei
ZH-1006	frei
ZH-1007	frei

licenseplate	freioderbelegt
ZH-1008	frei
ZH-1009	frei

54

- die Kennzeichen der Fahrzeuge zusammen mit der Anzahl der Reservationen am 11.11.2024 von 19-22 Uhr
- all license plates with the number of reservations free on 09.09.2024 from 19-22
- Difficulty: Difficult (++++)
- Hint: Predicates with date/time values, plus grouping and aggregation

```
select licensePlate,
       (select count(resnumber)
        from reservation r
        where r.vehicle = v.id
          and tsrange((r.interval).begints, (r.interval).endts, '[')
                && tsrange('2024-11-11 19:00:00', '2024-11-11 22:00:00', '[')) as rescnt
from vehicle v
order by rescnt desc, licenseplate;
```

Displaying records 1 - 10

licenseplate	rescnt
194 SUA	2
ZH-1001	1
ZH-1002	1
ZH-1003	1
ZH-1004	1
ZH-1005	1
ZH-1006	1
ZH-1007	1
ZH-1008	1
ZH-1009	1

55

- die Kuerzel der Stationen zusammen mit der Anzahl der Fahrzeuge, die am 11.11.2024 von 9-10 Uhr frei sind
- the short names of locations with the number of vehicles at each location which are free on 11.11.2024 from 9-10
- Difficulty: Difficult (++++)
- Hint: Predicates with date/time values, plus grouping and aggregation

```
select shortName, name,
       (select count(*)
        from vehicle v
        where v.home = l.id
          and not exists ( select resnumber
                          from reservation r
                          where r.vehicle = v.id
                            and tsrange((r.interval).begints, (r.interval).endts, '[')
                                  && tsrange('2024-11-11 09:00:00', '2024-11-11 10:00:00', '['))
        ) as vehicleCount
from location l
order by 3 desc;
```

Displaying records 1 - 10

shortname	name	vehiclecount
zrh	Zurich	8
swb	Schwerzenbach	7
grs	Greifensee	5
ust	Uster	5
irc	Zurich-Irchel	5
moe	Moenchaltorf	4
wip	Zurich-Wipkingen	3
see	Zurich-Seefeld	3
oer	Zurich-Oerlikon	2
lei	Zurich-Leimbach	1

56

- Benutzer ohne Reservationen
- members without any reservations
- Difficulty: Moderate (+++)
- Hint: There are multiple wazys ... one is to use the right type of join and then test for ... nothing

```
select m.id
from member m left outer join reservation r on r.member = m.id
where r.id is null;
```

8 records

id

	id
	32498
	32496
	32497
	1033
	32499
	1026
	32500
	1010

57

- die Kennzeichen und Anzahl der Reservationen der einzelnen Fahrzeuge
- the license plates and number of reservations of each vehicle
- Difficulty: Moderate (+++)
- Hint: Joins should be familiar by now. Here you also need grouping and aggregation

```
select licensePlate, count(distinct r.id)
  from vehicle v left outer join
    reservation r on r.vehicle = v.id
 group by licenseplate
 order by 1;
```

Displaying records 1 - 10

licenseplate	count
194 SUA	171
ZH-1001	39
ZH-1002	48
ZH-1003	48
ZH-1004	48
ZH-1005	48
ZH-1006	45
ZH-1007	60
ZH-1008	24
ZH-1009	47

58

- der hoechste Kilometerstand eines Fahrzeugs
- the maximum odometer value
- Difficulty: Easy (++)
- Hint: Choose the right aggregation operator

```
select max(kilometers)
  from Vehicle;
```

1 records

max
185333

59

- das Fahrzeug mit dem hoechsten Kilometerstand
- the vehicle with the highest odometer value
- Difficulty: Moderate (+++)
- Hint: Reuse what you did in the previous query!

```
select licensePlate
  from Vehicle
 where kilometers = (select max(kilometers)
                    from Vehicle);
```

1 records

licenseplate
194 SUA

60

- der hoechste Benzinverbrauch eines Fahrzeugs
- the highest fuel consumption of all vehicles
- Difficulty: Easy (++)
- Hint: Choose the right aggregation operator

```
select max(gasconsumption)
  from Vehicle;
```

1 records

max

max

54702.32

61

- das Fahrzeug mit dem höchsten Benzinverbrauch
- the vehicle with the highest fuel consumption
- Difficulty: Moderate (+++)
- Hint: Reuse what you did in the previous query!

```
select licensePlate
  from Vehicle
 where gasconsumption = (select max(gasconsumption)
                        from Vehicle);
```

1 records

licenseplate
194 SUA

194 SUA

62

- die Anzahl Limousinen pro Station
- the number of limousines per location
- Difficulty: Moderate (+++)
- Hint: You need several operators, each of which should be well-known by now (joins, grouping, aggregation, ...)

```
select shortname, count(v.id)
  from limousine v right outer join location l on v.home = l.id
 group by shortname;
```

Displaying records 1 - 10

shortname	count
oer	1
zrh	2
swb	0
ust	1
irc	0
lin	0
moe	1
grs	1
wip	0
lei	0

63

- die maximale Anzahl Limousinen an einer Station
- the highest number of limousines per location
- Difficulty: Difficult (++++)
- Hint: Reuse what you did in the previous query!

```
select max(limocnt) as maxlimo
  from (select shortname, count(v.id) as limocnt
        from limousine v right outer join location l on v.home = l.id
        group by shortname) as lc;
```

1 records

maxlimo
2

2

64a

- Die Station mit der maximalen Anzahl Limos
- the location with the highest number of limousines
- Difficulty: Difficult (++++)
- Hint: Reuse what you did in the previous query!

```
select shortname, count(v.id) as limocnt
  from limousine v right outer join location l on v.home = l.id
 group by shortname
 having count(v.id) = (select max(limocnt) as maxlimo
                    from (select shortname, count(v.id) as limocnt
                          from limousine v right outer join location l on v.home = l.id
                          group by shortname) as lc);
```

1 records

shortname	limocnt
zrh	2

64

- Die Station mit der maximalen Anzahl Limos
- the location with the highest number of limousines
- Difficulty: Difficult (++++)

- Hint: Reuse what you did in the previous query!

```
with limos(shortName, limoCnt) as
  (select shortname, count(v.id)
   from limousine v right outer join location l on v.home = l.id
   group by shortname)
select shortName, limocnt
from limos
where limoCnt = (select max(limocnt) from limos);
```

1 records

shortname	limocnt
zrh	2

65a

- alle Stationen, die (ueber die Nachbarschaftsbeziehung) in maximal vier Schritten von Mönchaltorf (moe) erreichbar sind
- all locations that are reachable (via the nextLocation relationship) in no more than four steps from location moe
- Difficulty: Very difficult (++++)
- Hint: A typical case for recursive queries

```
update location
set nextLocation = 1005, secondNextLocation = 1006
where shortName = 'grs';
```

65b

- alle Stationen, die (ueber die Nachbarschaftsbeziehung) in maximal vier Schritten von Mönchaltorf (moe) erreichbar sind
- all locations that are reachable (via the nextLocation relationship) in no more than four steps from location moe
- Difficulty: Very difficult (++++)
- Hint: A typical case for recursive queries

```
update location
set nextLocation = 1004, secondNextLocation = 1000
where shortName = 'swb';
```

65

- alle Stationen, die (ueber die Nachbarschaftsbeziehung) in maximal vier Schritten von Mönchaltorf (moe) erreichbar sind
- all locations that are reachable (via the nextLocation relationship) in no more than four steps from location moe
- Difficulty: Very difficult (++++)
- Hint: A typical case for recursive queries

```
with recursive neighbors(shortName, next, steps) as
  ((select l.shortName,
    n.shortName as next,
    1
   from location l join
    location n on l.nextLocation = n.id
   where l.shortName = 'moe')
 union all
  (select l.shortName,
    l2.shortName as next,
    steps + 1
   from location l join
    neighbors n on l.shortName = n.next join
    location l2 on l.nextLocation = l2.id
   where l2.shortName <> 'moe'
    and steps <= 3 ))
select shortName, next, steps
from neighbors
order by steps, shortName, next;
```

4 records

shortname	next	steps
moe	ust	1
ust	grs	2
grs	swb	3
swb	see	4

66

- der durchschnittliche Spritverbrauch pro Fahrzeug
- the average fuel consumption (per 100km) per vehicle
- Difficulty: Easy (++)
- Hint: beware of division by 0!

```
select licensePlate,
  to_char(gasConsumption / (kilometers / 100.0), '99D99') as averageGasConsumption
from vehicle
where kilometers > 0;
```

Displaying records 1 - 10

licenseplate	averagegasconsumption
ZH-1004	5.22
ZH-1005	5.22
ZH-1015	4.92

licenseplate	averagegasconsumption
ZH-1016	5.31
ZH-8203	5.33
ZH-4001	5.50
ZH-4005	4.50
ZH-4006	4.50
ZH-4007	4.50
ZH-4008	4.50

67

- der maximale durchschnittliche Spritverbrauch pro Fahrzeug
- the highest average fuel consumption per vehicle
- Difficulty: Moderate (+++)
- Hint: Reuse the previous query

```
select max(to_char(gasConsumption / (kilometers / 100.0), '99D99')) as maxAverageGasConsumption
from vehicle
where kilometers > 0;
```

1 records

maxaveragegasconsumption
9.08

68

- das Fahrzeug mit dem maximalen Spritverbrauch
- the vehicle with the highest average fuel consumption
- Difficulty: Moderate (+++)
- Hint: Reuse the previous query

```
with gas(car, gasUsage) as
(select licensePlate,
to_char(gasConsumption / (kilometers / 100.0), '99D99') as averageGasConsumption
from vehicle
where kilometers > 0)
select car, gasUsage
from gas
where gasUsage = (select max(gasUsage) from gas);
```

1 records

car	gasusage
ZH-1006	9.08

69

- der maximale, minimale und durchschnittliche Spritverbrauch pro Fahrzeugtyp
- the highest, average, and smallest average fuel consumption by vehicle category
- Difficulty: Easy (++)
- Hint: just multiple aggregation operators in one select

```
select type,
to_char(min(gasConsumption / (kilometers / 100.0)), '99D99') as minGasConsumption,
to_char(avg(gasConsumption / (kilometers / 100.0)), '99D99') as avgGasConsumption,
to_char(max(gasConsumption / (kilometers / 100.0)), '99D99') as maxGasConsumption
from vehicle
where kilometers > 0
group by type
order by type ;
```

8 records

type	mingasconsumption	avvgasconsumption	maxgasconsumption
CompactCar	4.50	4.91	5.50
Convertible	10.26	10.60	11.00
Limousine	13.01	16.33	29.52
Pickup	16.55	17.17	17.75
Sedan	7.06	8.12	8.69
StationWagon	8.53	9.10	10.25
Transporter	13.06	13.23	13.40
Van	12.13	12.96	13.98

70

- maximaler Spritverbrauch pro Fahrzeug und Benutzer
- the highest fuel consumption by member and vehicle (for a single reservation)
- Difficulty: Moderate (+++)
- Hint: You can have more than one grouping variable

```
select max(cast(gas as decimal(10,2)))
  from (select member, vehicle,
            100 * avg(cast(gasConsumed as decimal(10,2))
                    / cast(kilometers as decimal(10,2))) as gas
        from useofvehicle u join
            reservation r on u.reservation = r.id
        where kilometers > 0
        group by member, vehicle ) as gu;
```

1 records

	max
	22.84

71

- der Benutzer und das Fahrzeug mit dem höchsten Benzinverbrauch (für eine einzelne Benutzung)
- member and vehicle with the highest fuel consumption (for a single reservation)
- Difficulty: Difficult (++++)
- Hint: Reuse the previous query

```
with gasUse(member, car, gas) as
  (select member, vehicle,
    gasconsumed / (kilometers / 100.0) as gas
   from useofvehicle u join
       reservation r on u.reservation = r.id
   where kilometers > 0)
select *
  from gasUse gu
 where gu.gas = (select max(gas) from gasuse);
```

1 records

member	car	gas
1021	1000	33.76923

72

- das Fahrzeug mit dem maximalen Spritverbrauch
- the vehicle with the highest fuel consumption (NOT per kilometer)
- Difficulty: Moderate (+++)
- Hint: Reuse one of the previous queries

```
with gas(gasUsage) as
  (select max(cast(gasConsumption as decimal(10,2))) as gasUsage
   from vehicle)
select licensePlate, gasUsage
  from gas, vehicle
 where gasUsage = gasConsumption;
```

1 records

licenseplate	gasusage
194 SUA	54702.32

73

- Kategorien und der höchste Kilometerstand pro Kategorie
- Vehicle categories and the highest odometer value for each category
- Difficulty: Moderate (+++)
- Hint: Another case for grouping and aggregation

```
select type, max(kilometers)
  from vehicle
 group by type
 order by type;
```

8 records

type	max
CompactCar	33472
Convertible	27076
Limousine	185333
Pickup	57227
Sedan	77984
StationWagon	60495
Transporter	37151
Van	45348

74

- Kategorien und der höchste Kilometerstand pro Kategorie sowie das Kennzeichen des Fahrzeugs (dieser Kategorie) mit diesem Kilometerstand
- Vehicle categories and the highest odometer value for each category, plus the license plate of the vehicle with this odometer value
- Difficulty: Difficult (++++)
- Hint: Compute the maximum values and then find the vehicles with these values (note that joins can have more than one condition)


```

with tmp as (
select type, max(kilometers) as maxkm
  from vehicle v
 group by type)
select v.type, maxkm, licenseplate
  from vehicle v join
      tmp t on t.type = v.type and v.kilometers = t.maxkm
 order by v.type;

```

8 records

type	maxkm	licenseplate
CompactCar	33472	ZH-8203
Convertible	27076	ZH-8204
Limousine	185333	194 SUA
Pickup	57227	ZH-8207
Sedan	77984	ZH-8201
StationWagon	60495	ZH-1006
Transporter	37151	ZH-8208
Van	45348	ZH-8206

75

- Die Benutzungsdauern der Reservationen
- the duration of the reservations
- Difficulty: Easy (++)
- Hint: Date/time arithmetic

```
select hours from useofvehicle;
```

Displaying records 1 - 10

hours
8
8
24
8
20
49
8
24
9
8

76

- Die Benutzungsdauern der Reservationen (jeden Wert nur einmal)
- the duration of the reservations, list each value only once
- Difficulty: Easy (++)
- Hint:

```
select distinct hours from useofvehicle;
```

Displaying records 1 - 10

hours
11
24
8
19
4
30
0
40
49
3

77

- Die Benutzungsdauern der Reservationen (jeden Wert nur einmal), die längste Benutzung zuerst
- the duration of the reservations, list each value only once and sort from largest to smallest
- Difficulty: Moderate (+++)
- Hint:

```
select distinct hours
  from useofvehicle
 order by hours desc;
```

Displaying records 1 - 10

	hours
	100
	49
	48
	40
	30
	24
	22
	20
	19
	18

78

- Ein Histogramm der Benutzungsdauern der Reservationen, d.h. zu jeder Dauer die Angabe, wie oft sie vorkommt
- a histogram of reservation durations, i.e., for each duration calculate how often it occurs
- Difficulty: Difficult (++++)
- Hint:

```
select hours, count(*) as cnt
from useofvehic1e
group by hours
order by hours desc;
```

Displaying records 1 - 10

	hours	cnt
	100	6
	49	5
	48	11
	40	18
	30	6
	24	99
	22	12
	20	175
	19	6
	18	4

79

- die Kennzeichen und Anzahl der Reservationen der einzelnen Fahrzeuge, absteigend sortiert nach Anzahl der Reservationen, zusammen mit der kummulierten Summe der Reservationen und dem Anteil des Fahrzeugs an dieser kummulierten Summe (kummulierte Summe = Summe der Reservationen aller Fahrzeuge bis einschliesslich des aktuellen in der Liste)
- the license plates and number of reservations per vehicle. order should be by number of reservations (largest to smallest). Also compute the cumulated sum of the reservations and the contribution of the current vehicle to the cumulated sum
- Difficulty: Difficult (++++)
- Hint: Use the right analytic function in the select clause

```
select licensePlate,
resCnt,
sum(resCnt) over (order by resCnt desc
                  ROWS unbounded preceding) as cumulatedCnt,
(100.0 * resCnt) / sum(resCnt) over(order by resCnt desc
                                   ROWS between unbounded preceding
                                   and unbounded following)
as fraction
from (select licensePlate, count(r.id) as resCnt
      from reservation r right outer join
           vehicle v on r.vehicle = v.id
      group by licensePlate) as tmp
order by resCnt desc;
```

Displaying records 1 - 10

licenseplate	rescnt	cumulatedcnt	fraction
194 SUA	171	171	7.833257
ZH-8201	96	267	4.397618
ZH-8207	93	360	4.260192
ZH-8204	84	444	3.847916
ZH-1019	72	516	3.298214
ZH-3462	72	588	3.298214
ZH-3463	72	660	3.298214
ZH-1031	69	729	3.160788
ZH-1028	60	789	2.748511
ZH-1017	60	849	2.748511

80

- Jahr und Monat der Reservationen (Reservationsbeginn, jede Kombination nur einmal)
- year and month of reservations (reservation begin; each combination only once)
- Difficulty: Easy (++)
- Hint: Once more date/time comparisons

```
select distinct extract(year from (interval).begints) as jahr,
               extract(month from (interval).begints) as Monat
  from reservation
 order by jahr, monat;
```

Displaying records 1 - 10

jahr	monat
2024	6
2024	7
2024	8
2024	9
2024	10
2024	11
2024	12
2025	1
2025	2
2025	3

81

- Jahr und Monat der Reservationen und Anzahl der Reservationen in diesem Monat (ausschlaggebend ist der Reservationsbeginn)
- year and month of reservations and number of reservations during this month (use reservation begin)
- Difficulty: Moderate (+++)
- Hint: Use the previous query and add a group operator

```
select extract(year from (interval).begints) as jahr,
       extract(month from (interval).begints) as Monat,
       count(*) as cnt
  from reservation
 group by extract(year from (interval).begints),
          extract(month from (interval).begints)
 order by 1, 2;
```

Displaying records 1 - 10

jahr	monat	cnt
2024	6	2
2024	7	58
2024	8	174
2024	9	182
2024	10	183
2024	11	183
2024	12	174
2025	1	183
2025	2	178
2025	3	169

82

- Jahr und Monat der Reservationen und Summe der Reservationsdauern in diesem Monat (ausschlaggebend ist der Reservationsbeginn)
- year and month of reservations and sum of reservation durations during this month (use reservation begin)
- Difficulty: Moderate (+++)
- Hint: Use the previous query and add a group operator

```
select extract(year from (interval).begints) as jahr,
       extract(month from (interval).begints) as Monat,
       sum(hours) as cnt
  from reservation r join useofvehicle u on u.reservation = r.id
 group by extract(year from (interval).begints),
          extract(month from (interval).begints)
 order by 1, 2;
```

8 records

jahr	monat	cnt
2024	6	8
2024	7	1054
2024	8	2122
2024	9	2426
2024	10	2446
2024	11	2400
2024	12	2264

jahr	monat	cnt
2025	1	1779

83a

- Jahr und Monat der Reservationen und Anzahl der Reservationen in diesem Monat (ausschlaggebend ist der Reservationsbeginn); und Bildung einer Rangliste innerhalb der einzelnen Jahre
- year and month of reservations and number of reservations during this month (use reservation begin). Sort according to number of reservations within each year
- Difficulty: Difficult (++++)
- Hint: Use the appropriate analytic function in the select clause

```
select jahr, monat, cnt,
       rank() over(partition by jahr order by cnt desc) as rang
from (select extract(year from (interval).begints) as jahr,
             extract(month from (interval).begints) as Monat,
             count(*) as cnt
      from reservation
      group by extract(year from (interval).begints), extract(month from (interval).begints)) t
order by jahr, rang;
```

Displaying records 1 - 10

jahr	monat	cnt	rang
2024	10	183	1
2024	11	183	1
2024	9	182	3
2024	12	174	4
2024	8	174	4
2024	7	58	6
2024	6	2	7
2025	6	184	1
2025	1	183	2
2025	4	181	3

83

- Jahr und Monat der Reservationen und Anzahl der Reservationen in diesem Monat (ausschlaggebend ist der Reservationsbeginn); und Bildung einer Rangliste innerhalb der einzelnen Jahre
- year and month of reservations and number of reservations during this month (use reservation begin). Sort according to number of reservations within each year
- Difficulty: Difficult (++++)
- Hint: Use the appropriate analytic function in the select clause

```
with ResProMonat as
  (select extract(year from (interval).begints) as jahr,
         extract(month from (interval).begints) as Monat,
         count(*) as cnt
   from reservation
   group by extract(year from (interval).begints), extract(month from (interval).begints))
select jahr, monat, cnt,
       rank() over(partition by jahr order by cnt desc) as rang
from ResProMonat
order by jahr, rang;
```

Displaying records 1 - 10

jahr	monat	cnt	rang
2024	10	183	1
2024	11	183	1
2024	9	182	3
2024	12	174	4
2024	8	174	4
2024	7	58	6
2024	6	2	7
2025	6	184	1
2025	1	183	2
2025	4	181	3

84

- gleich wie oben, Summe der Reservationsdauern anstatt Anzahl Reservationen
- same as above, but use sum of reservation durations instead of reservation count
- Difficulty: Difficult (++++)
- Hint: Use the previous query and add a group operator

85

- Jahr und Monat der Reservationen und Summe der Reservationsdauern in diesem Monat (ausschlaggebend ist der Reservationsbeginn); und Bildung einer Rangliste innerhalb der einzelnen Jahre. Ausgabe der auf der Rangliste führenden beiden Monate
- same as above, but use sum of reservation durations instead of reservation count. Return only the first two months
- Difficulty: Difficult (++++)

- Hint: Use the appropriate analytic function in the select clause. Use nested queries if needs be

```
select *
from (select jahr, monat, totalhours,
      rank() over(partition by jahr order by totalhours desc) as rang
      from (select extract(year from (interval).begints) as jahr,
                extract(month from (interval).begints) as Monat,
                sum(hours) as totalhours
            from reservation r join useofvehicle u on u.reservation = r.id
            group by extract(year from (interval).begints), extract(month from (interval).begints)) t1) t2
where rang < 3
order by jahr, rang;
```

3 records

jahr	monat	totalhours	rang
2024	10	2446	1
2024	9	2426	2
2025	1	1779	1

86

- gleitender 3-Monatsdurchschnitt der Reservationsanzahlen
- calculate the 3-months moving average of reservation counts
- Difficulty: Difficult (++++)
- Hint: Another example for analytic operators

```
select jahr, monat, cnt,
      avg(cnt) over(partition by jahr
                  order by monat
                  rows between 1 preceding and 1 following) as mvgavg
from (select extract(year from (interval).begints) as jahr,
            extract(month from (interval).begints) as Monat,
            count(*) as cnt
     from reservation
     group by extract(year from (interval).begints),
            extract(month from (interval).begints)) t
order by jahr, monat;
```

Displaying records 1 - 10

jahr	monat	cnt	mvgavg
2024	6	2	30.0000
2024	7	58	78.0000
2024	8	174	138.0000
2024	9	182	179.6667
2024	10	183	182.6667
2024	11	183	180.0000
2024	12	174	178.5000
2025	1	183	180.5000
2025	2	178	176.6667
2025	3	169	176.0000

87

- monatsweise kummulierte Summe der Reservationszahlen innerhalb eines Jahres
- calculate the cumulated sums of monthly reservation counts per year
- Difficulty: Difficult (++++)
- Hint: Another example for analytic operators

```
select jahr, monat, cnt,
      sum(cnt) over(partition by jahr order by monat rows unbounded preceding ) as cumsum
from (select extract(year from (interval).begints) as jahr,
            extract(month from (interval).begints) as Monat,
            count(*) as cnt
     from reservation
     group by extract(year from (interval).begints), extract(month from (interval).begints)) t
order by jahr, monat;
```

Displaying records 1 - 10

jahr	monat	cnt	cumsum
2024	6	2	2
2024	7	58	60
2024	8	174	234
2024	9	182	416
2024	10	183	599
2024	11	183	782
2024	12	174	956
2025	1	183	183
2025	2	178	361
2025	3	169	530

88

- Einteilung der Monate des Jahres 2024 in Tertile (drei Quantile) gemäss der Anzahl Reservationen
- assign the months of the year 2024 to tertiles (three quantiles) based on the number of reservations per month
- Difficulty: Difficult (++++)
- Hint: Another example for analytic operators

```
select jahr, monat, cnt,
       ntile(3) over(partition by jahr order by cnt ) as tiles
from (select extract(year from (interval).begints) as jahr,
            extract(month from (interval).begints) as Monat,
            count(*) as cnt
      from reservation
      group by extract(year from (interval).begints), extract(month from (interval).begints)) t
where jahr = 2024;
```

7 records

jahr	monat	cnt	tiles
2024	6	2	1
2024	7	58	1
2024	12	174	1
2024	8	174	2
2024	9	182	2
2024	11	183	3
2024	10	183	3

89

- Anzahl Reservationen gruppiert nach den Dimensionen Mitglied, Fahrzeug, Station
- The number of reservations grouped over the dimensions member, vehicle, and location
- Difficulty: Difficult (++++)
- Hint: Super groups: grouping sets

```
select membernr, v.licenseplate, l.name, count(*)
from reservation r join
     vehicle v on r.vehicle = v.id join
     member m on r.member = m.id join
     location l on v.home = l.id
group by grouping sets (membernr, v.licenseplate, l.name, ())
Order by 1,2,3;
```

Displaying records 1 - 10

membernr	licenseplate	name	count
1000	NA	NA	124
1001	NA	NA	121
1003	NA	NA	83
1004	NA	NA	96
1005	NA	NA	84
1006	NA	NA	72
1007	NA	NA	96
1008	NA	NA	73
1009	NA	NA	48
1010	NA	NA	95

90

- Anzahl Reservationen gruppiert nach allen möglichen Kombinationen der Dimensionen Mitglied und Fahrzeugtyp
- The number of reservations grouped over all possible combinations of the dimensions member (member number) and vehicle type.
- Difficulty: Difficult (++++)
- Hint: Super groups: cube

```
select membernr, v.type, count(*)
from reservation r join
     vehicle v on r.vehicle = v.id join
     member m on r.member = m.id
group by cube (membernr, v.type)
Order by 1,2;
```

Displaying records 1 - 10

membernr	type	count
1000	Convertible	12
1000	Limousine	51
1000	Pickup	24
1000	Sedan	1
1000	StationWagon	12
1000	Van	24
1000	NA	124
1001	Convertible	24

membernr	type	count
1001	Limousine	71
1001	Pickup	24

91

- Anzahl Reservationen gruppiert nach allen möglichen Kombinationen der Dimensionen Mitglied und Fahrzeugtyp, so dass auch Mitglieder und Fahrzeugtypen ohne Reservationen aufgeführt werden
- The number of reservations grouped over all possible combinations of the dimensions member (member number) and vehicle type such that also types and members without reservations are shown.
- Difficulty: Difficult (++++)
- Hint: see previous query

```
select membernr, v.type, count(resnumber)
  from reservation r full outer join
    vehicle v on r.vehicle = v.id full join
    member m on r.member = m.id
 group by cube (membernr, v.type)
 order by 1,2;
```

Displaying records 1 - 10

membernr	type	count
1000	Convertible	12
1000	Limousine	51
1000	Pickup	24
1000	Sedan	1
1000	StationWagon	12
1000	Van	24
1000	NA	124
1001	Convertible	24
1001	Limousine	71
1001	Pickup	24

92

- Anzahl Reservationen gruppiert nach allen möglichen Kombinationen der Dimensionen Mitglied und Fahrzeugtyp, so dass auch Mitglieder und Fahrzeugtypen ohne Reservationen aufgeführt werden. Ausserdem sollen All- und Nullwerte unterschieden werden.
- The number of reservations grouped over all possible combinations of the dimensions member (member number) and vehicle type such that also types and members without reservations are shown. In addition distinguish all and null values.
- Difficulty: Difficult (++++)
- Hint: see previous query

```
select case when grouping(membernr)=1 then 'all' else membernr::text end as member,
       case when grouping(v.type)=1 then 'all' else v.type end as type,
       count(resnumber)
  from reservation r full outer join
    vehicle v on r.vehicle = v.id full join
    member m on r.member = m.id
 group by cube (membernr, v.type)
 order by 1,2;
```

Displaying records 1 - 10

member	type	count
1000	all	124
1000	Convertible	12
1000	Limousine	51
1000	Pickup	24
1000	Sedan	1
1000	StationWagon	12
1000	Van	24
1001	all	121
1001	Convertible	24
1001	Limousine	71

93

- Anzahl Reservationen hierarchisch gruppiert nach Jahr, Monat und Tag des Reservationsbeginns
- The number of reservations hierarchically grouped over year, month, and day of the reservation begin
- Difficulty: Difficult (++++)
- Hint: Super groups: rollup

```
select extract(year from (r.interval).begints) as year,
       to_char((r.interval).begints, 'YYYY-MM') as month,
       date_trunc('day', (r.interval).begints)::date as day,
       count(resnumber)
  from reservation r
 group by rollup(extract(year from (r.interval).begints),
                to_char((r.interval).begints, 'YYYY-MM'),
                date_trunc('day', (r.interval).begints))
 order by 1,2,3;
```

Displaying records 1 - 10

year	month	day	count
2024	2024-06	2024-06-26	2
2024	2024-06	NA	2
2024	2024-07	2024-07-04	1
2024	2024-07	2024-07-07	1
2024	2024-07	2024-07-08	1
2024	2024-07	2024-07-12	8
2024	2024-07	2024-07-13	1
2024	2024-07	2024-07-14	3
2024	2024-07	2024-07-16	4
2024	2024-07	2024-07-17	3

94

- alle Benutzer, die fern der Heimat reserviert haben (Fahrzeugstandort <> Heimatstandort des Benutzers)
- all members who have reserved a vehicle at a location other than their home location
- Difficulty: Easy (++)
- Hint:

```
select m.memberNr
from reservation r join
    member m on r.member = m.id join
    vehicle v on r.vehicle = v.id
where m.homelocation <> v.home;
```

Displaying records 1 - 10

membernr
1001
1001
1000
1004
1010
1008
1007
1007
1007
1007

95

- alle Benutzer, die fern der Heimat reserviert haben (Fahrzeugstandort <> Heimatstandort des Benutzers), obwohl es den Fahrzeugtyp an der Heimatstation gibt
- all members who have reserved a vehicle at a location other than their home location even though the vehicle type exists at their home location
- Difficulty: Difficult (++++)
- Hint: Subquery in where-clause

```
select m.memberNr, r.resNumber
from reservation r join
    member m on r.member = m.id join
    vehicle v on r.vehicle = v.id
where m.homelocation <> v.home
    and (select count(*)
        from vehicle c
        where c.home = m.homelocation
            and c.type = v.type) > 0;
```

Displaying records 1 - 10

membernr	resnumber
1010	54
1010	2067
1010	1884
1010	1701
1010	1518
1010	1335
1010	1152
1010	969
1010	786
1010	603

96

- alle Benutzer, die fern der Heimat reserviert haben (Fahrzeugstandort <> Heimatstandort des Benutzers), obwohl es den Fahrzeugtyp an der Heimatstation gibt und eines der Fahrzeuge des gewünschten Typs frei war/wäre.

- all members who have reserved a vehicle at a location other than their home location even though the vehicle type exists at their home location and a vehicle of the desired time would be free during the interval in question
- Difficulty: Difficult (++++)
- Hint: Subqueries in where-clause

```

select m.memberNr, r.resNumber
  from reservation r join
    member m on r.member = m.id join
    vehicle v on r.vehicle = v.id
 where m.homelocation <> v.home
    and (select count(*)
         from vehicle c
        where c.home = m.homelocation
          and c.type = v.type
          and not exists(select resnumber
                        from reservation r2
                       where ( (r.interval).begints <= (r2.interval).begints
                             and (r.interval).endts > (r2.interval).begints)
                          Or
                          ( (r.interval).begints > (r2.interval).begints
                             and (r.interval).begints <= (r2.interval).endts))
                        ) > 0;

```

0 records

membernrresnumber

97

- ein Belegungsplan fuer das Auto '194 SUA' fuer den 11.11. 2024 und die folgenden beiden Tage
- a free/busy plan for the vehicle '194 SUA' for 11.11. 2024 and the following two days
- Difficulty: Very difficult (+++++)
- Hint: Procedurally, you would use a loop. In SQL you can do the same with recursion

```

with recursive resTable(license, anfang, ende, isFree, ind) as
  ((select '194 SUA',
    cast('2024-11-11 00:00:00' as Timestamp),
    cast('2024-11-11 01:00:00' as Timestamp),
    not exists ( select resnumber
                from reservation r
                where r.vehicle = v.id
                  and ( ( (r.interval).begints <= '2024-11-11 00:00:00'
                        and (r.interval).endts > '2024-11-11 00:00:00')
                    or
                    ( (r.interval).begints > '2024-11-11 01:00:00'
                      and (r.interval).begints <= '2024-11-11 01:00:00'))),
    1
   from   vehicle v
   where  licensePlate = '194 SUA')
 union all
  (select r.license,
    anfang + '1 HOUR',
    ende + '1 HOUR',
    not exists ( select resnumber
                from reservation r
                where r.vehicle = v.id
                  and ( ( (r.interval).begints <= anfang + '1 HOUR'
                        and (r.interval).endts > anfang + '1 HOUR')
                    or
                    ( (r.interval).begints > ende + '1 HOUR'
                      and (r.interval).begints <= ende + '1 HOUR'))),
    ind + 1
   from   resTable r, vehicle v
   where  r.license = v.licensePlate
          and ind < 72))
select license, anfang, ende, isFree
  from   resTable;

```

Displaying records 1 - 10

license	anfang	ende	isfree
194 SUA	2024-11-11 00:00:00	2024-11-11 01:00:00	TRUE
194 SUA	2024-11-11 01:00:00	2024-11-11 02:00:00	TRUE
194 SUA	2024-11-11 02:00:00	2024-11-11 03:00:00	TRUE
194 SUA	2024-11-11 03:00:00	2024-11-11 04:00:00	TRUE
194 SUA	2024-11-11 04:00:00	2024-11-11 05:00:00	TRUE
194 SUA	2024-11-11 05:00:00	2024-11-11 06:00:00	TRUE
194 SUA	2024-11-11 06:00:00	2024-11-11 07:00:00	TRUE
194 SUA	2024-11-11 07:00:00	2024-11-11 08:00:00	TRUE
194 SUA	2024-11-11 08:00:00	2024-11-11 09:00:00	TRUE
194 SUA	2024-11-11 09:00:00	2024-11-11 10:00:00	TRUE

98

- ein Belegungsplan fuer die Station Greifensee fuer den 11.11. 2024
- a free/busy plan for the location Greifensee for 11.11. 2024
- Difficulty: Very difficult (+++++)
- Hint: Procedurally, you would use a loop. In SQL you can do the same with recursion

```

with recursive resTable(location, license, anfang, ende, isFree, ind) as
((select shortname,
  licensePlate,
  cast('2024-11-11 00:00:00' as Timestamp),
  cast('2024-11-11 01:00:00' as Timestamp),
  not exists ( select resnumber
    from reservation r
    where r.vehicle = v.id
    and ( (r.interval).begints <= '2024-11-11 00:00:00'
    and (r.interval).endts > '2024-11-11 00:00:00')
    or
    ( (r.interval).begints > '2024-11-11 01:00:00'
    and (r.interval).begints <= '2024-11-11 01:00:00'))),
  1
  from vehicle v join
  location l on v.home = l.id
  where shortName = 'grs')
union all
(select location,
  r.license,
  anfang + '1 HOUR',
  ende + '1 HOUR',
  not exists ( select resnumber
    from reservation r
    where r.vehicle = v.id
    and ( (r.interval).begints <= anfang + '1 HOUR'
    and (r.interval).endts > anfang + '1 HOUR')
    or
    ( (r.interval).begints > ende + '1 HOUR'
    and (r.interval).begints <= ende + '1 HOUR'))),
  ind + 1
  from resTable r, vehicle v
  where r.license = v.licensePlate
  and ind < 48))
select location, license, anfang, ende, isFree, anfang
  from resTable
 order by location, anfang, license;

```

Displaying records 1 - 10

location	license	anfang	ende	isfree	anfang
grs	194 SUA	2024-11-11 00:00:00	2024-11-11 01:00:00	TRUE	2024-11-11 00:00:00
grs	ZH-3456	2024-11-11 00:00:00	2024-11-11 01:00:00	TRUE	2024-11-11 00:00:00
grs	ZH-3458	2024-11-11 00:00:00	2024-11-11 01:00:00	TRUE	2024-11-11 00:00:00
grs	ZH-3463	2024-11-11 00:00:00	2024-11-11 01:00:00	TRUE	2024-11-11 00:00:00
grs	ZH-4001	2024-11-11 00:00:00	2024-11-11 01:00:00	TRUE	2024-11-11 00:00:00
grs	194 SUA	2024-11-11 01:00:00	2024-11-11 02:00:00	TRUE	2024-11-11 01:00:00
grs	ZH-3456	2024-11-11 01:00:00	2024-11-11 02:00:00	TRUE	2024-11-11 01:00:00
grs	ZH-3458	2024-11-11 01:00:00	2024-11-11 02:00:00	TRUE	2024-11-11 01:00:00
grs	ZH-3463	2024-11-11 01:00:00	2024-11-11 02:00:00	TRUE	2024-11-11 01:00:00
grs	ZH-4001	2024-11-11 01:00:00	2024-11-11 02:00:00	TRUE	2024-11-11 01:00:00