

Datenbanken

Praktikum 3

Aufgabe 1

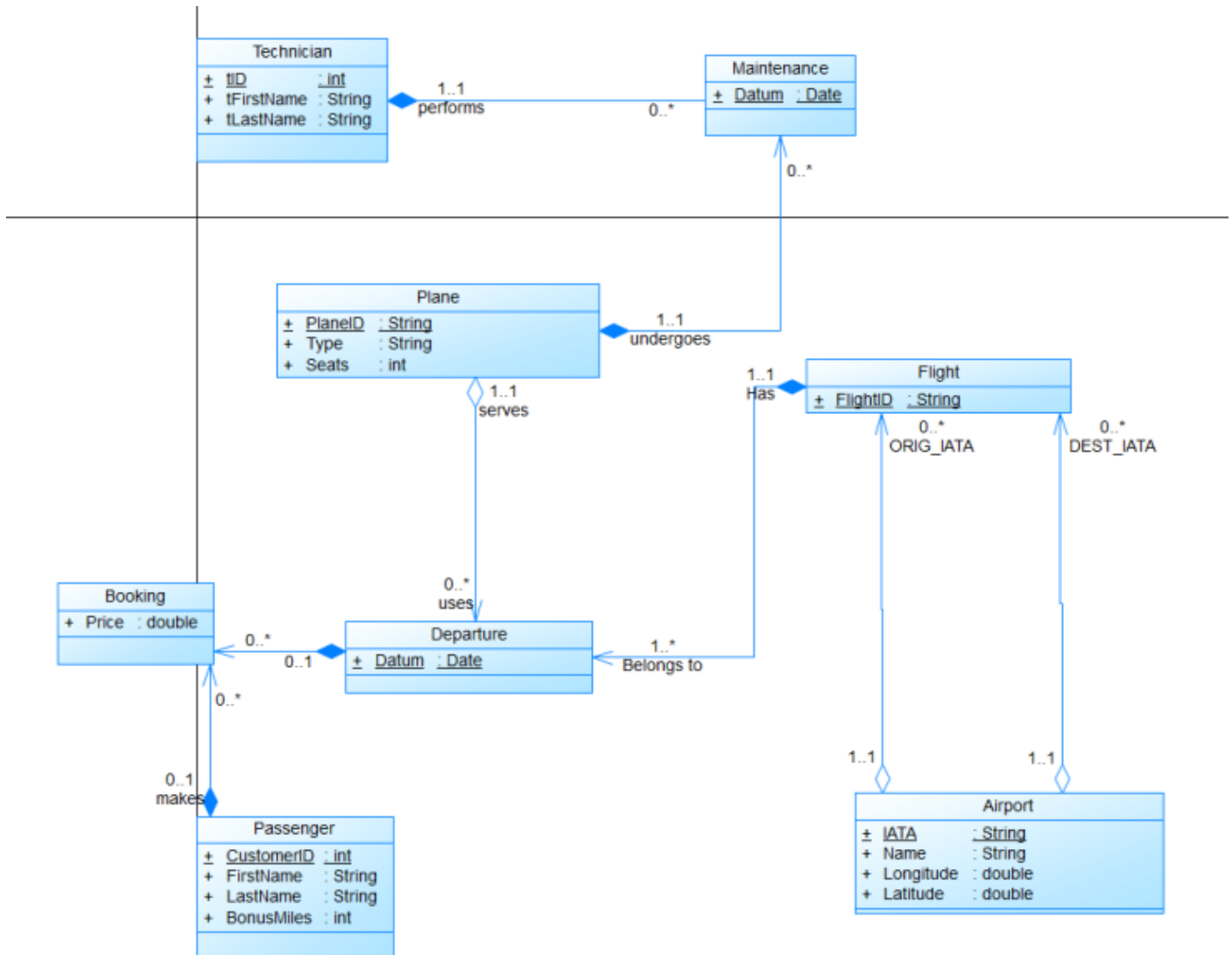


Bild 1, UML Klassendiagramm

Aufgabe 2

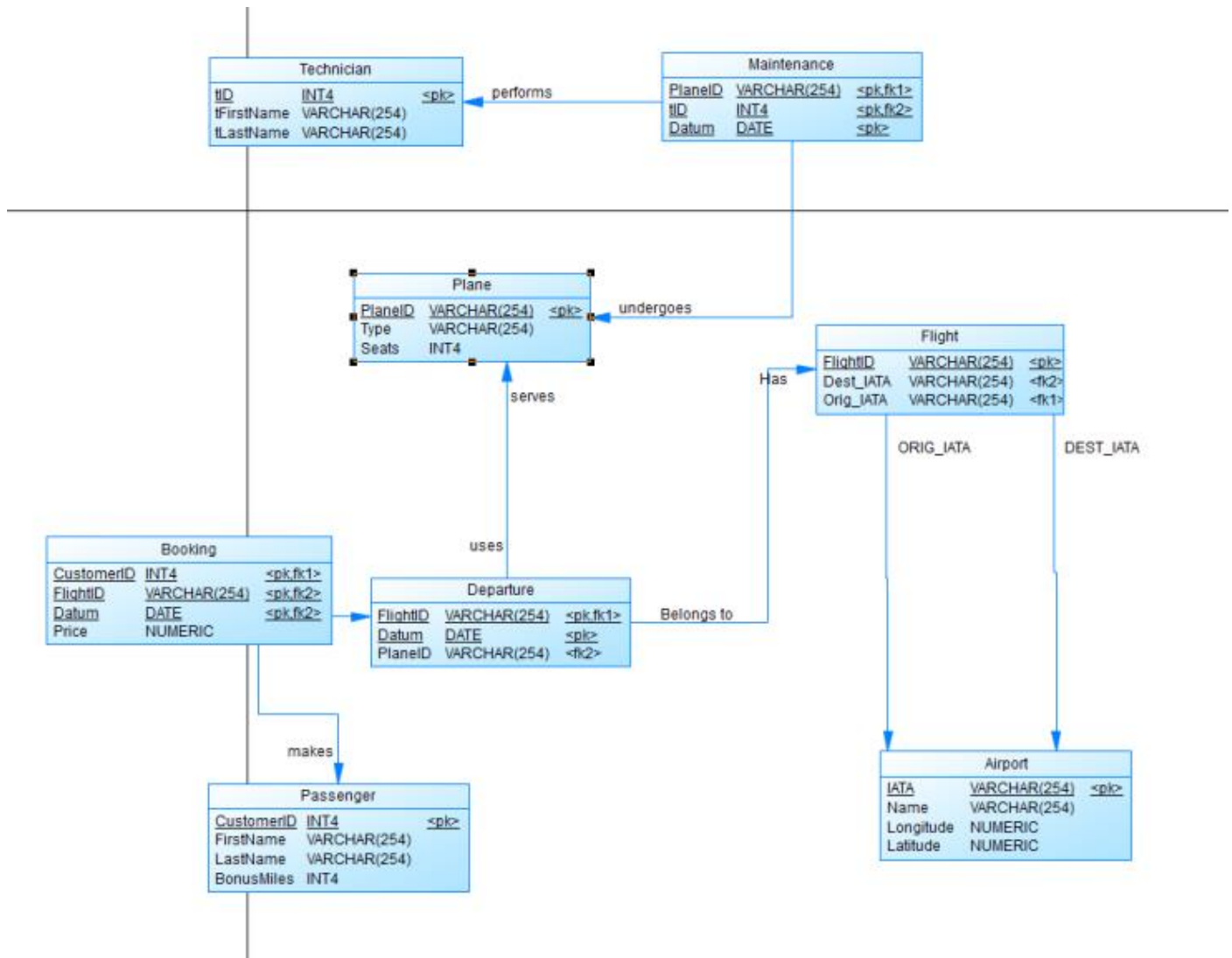


Bild 2, Physisches Modell

Aufgabe 3

3.1

Reihenfolge der „delete from TABELLENNAME“ spielt eine Rolle, das heißt, dass die Tabellen unter Voraussetzung gelöscht werden können. Falls Tabelle A die Tabelle B beinhaltet (Tabelle B benutzt Primary Key von Tabelle A als Fremd Key), kann Tabelle A erst gelöscht werden, wenn Tabelle B gelöscht ist, das heißt, wenn Primary Key von Tabelle A nicht mehr verwendet ist. Falls es versucht wird, eine Tabelle zu löschen, deren Primary Key eine andere Tabelle referenziert erscheint eine Fehlermeldung (siehe Bild 3 und Bild 4).

```
ERROR: update or delete on table "technician" violates foreign key constraint "fk_maintena_maintenan_technici" on table "maintenance"
DETAIL: Key (tid)=(1) is still referenced from table "maintenance".
SQL state: 23503
```

Bild 3, Fehlermeldung Beispiel 1

```
ERROR: update or delete on table "departure" violates foreign key constraint "fk_booking_departure_departur" on table "booking"
DETAIL: Key (flightid, datum)=(LH-100, 2018-01-10) is still referenced from table "booking".
SQL state: 23503
```

Bild 4, Fehlermeldung Beispiel 2

In unserem Fall, dürfen die Tabellen in folgenden Reihenfolge gelöscht werden (Hinweis, Reihenfolge kann auch anders sein, wichtig ist, dass „key constraint“ nicht verletzt ist).

Maintenance, Technician, Booking, Passenger, Departure, Plane, Flight, Airport.

Aufgabe 4

4.1

Airport

Primary Key

Name	Columns
pk_airport	iata

Foreign Key

Name	Columns	Referenced Table
/	/	/

Flight

Primary Key

Name	Columns
pk_flight	flightid

Foreign Key

Name	Columns	Referenced Table
fk_flight_flightdes_airport	(dest_iata) -> (iata)	airport
fk_flight_flightori_airport	(orig_iata) -> (iata)	airport

Tabelle Flight referenziert Tabelle Airport und zwar, zweimal. Einmal durch Destination IATA und einmal durch Origin IATA. Das bedeutet, dass auf Airport zwei Constraints liegen und die beide sind durch Aggregation entstanden.

Departure

Primary Key

Name	Columns
pk_departure	flightid,datum

Foreign Key

Name	Columns	Referenced Table
fk_departur_departure_flight	(flightid) -> (flightid)	flight
fk_departur_departure_plane	(planeid) -> (planeid)	plane

Departure Referenziert Tabellen Flight und Plane. Da zwischen Departure und Flight eine Komposition besteht, und zwar, Departure ist Teil von Flight, so ist Flight durch Kombination von Primary Key und Foreign Key <pk, fk1> referenziert (Kombination von Datum und FlightID). Zwischen Departure und Plane existiert eine Aggregation, und zwar, Departure verwendet („ist Teil von“) Plane, so wird Plane durch PlaneID als Fremd Key referenziert.

Plane

Primary Key

Name	Columns
pk_plane	planeid

Foreign Key

Name	Columns	Referenced Table
/	/	/

Technician

Primary Key

Name	Columns
pk_technician	tid

Foreign Key

Name	Columns	Referenced Table
/	/	/

Maintenance

Primary Key

Name	Columns
pk_maintenance	planeid,tid,datum

Foreign Key

Name	Columns	Referenced Table
fk_maintena_maintenan_technici	(tid) -> (tid)	technician
fk_maintena_planemain_plane	(planeid) -> (planeid)	plane

Maintenance referenziert zwei Tabellen, Technician und Plane. In beiden Fällen geht es um Komposition, das bedeutet, dass Maintenance sowohl Teil von Technician als auch von Plane ist. Technician ist durch Kombination von Primary Key und Foreign Key (tID) referenziert. Plane ist auch durch Kombination von Primary Key und Foreign Key (PlaneID). referenziert Das Primary Key von Maintenance ist durch Datum (Maintenance), PlaneID (Plane) und tID (Technician) als eine Kombination referenziert.

Passenger

Primary Key

Name	Columns
pk_passenger	customerid

Foreign Key

Name	Columns	Referenced Table
/	/	/

Boking

Primary Key

Name	Columns
pk_booking	flightid, customerid, datum

Foreign Key

Name	Columns	Referenced Table
fk_booking_departure_departur	(datum, flightid) -> (datum, flightid)	departure, departure
fk_booking_passenger_passenge	(customerid) -> (customerid)	passenger

Zwischen Booking und Departure besteht eine Komposition. Weil zwischen Departure und Flight auch eine Komposition besteht (Departure ist Teil von Flight), so wird Departure durch Tabelle Booking, zweimal referenziert, beide durch Kombination von Customer ID, Datum und Flight ID als Primary Key und Kombination von Flight ID und Datum als Fremd Key ,und da Power Designer (oder der Modell) kann eine Kombination von Objekten nicht zeigen, <pk, fk2> wird doppel als Datum und FightID separat gezeigt. Tabelle Passenger wird durch Kombination Customer ID, Flight ID und Datum und Customer ID als Fremd Key.

4.2

1) Einen Abflug für die Flugnummer „DB-123“ am 01.10.2018.

```
1 INSERT INTO Departure(PlaneID, FlightID, Datum) VALUES
2 ('D-ABBL', 'DB-123', '2018-01-10');
```

Data Output Messages Notifications

ERROR: insert or update on table "departure" violates foreign key constraint "fk_departur_departure_flight"
DETAIL: Key (flightid)=(DB-123) is not present in table "flight".
SQL state: 23503

Bild 5, Insert Departure + Fehlermeldung

Departure könnte in diesem Fall nicht hinzugefügt, weil in Tabelle Flight keine Flugnummer (FlightID) „DB-123“ gibt bzw. weil FlightID ein Fremd Key von Departure ist (Departure referenziert Tabelle Flight mit diesem Key), und dieses Key nicht in der Tabelle Flight ist.

2) Einen Flug von IATA-Code „FRA“ nach „ABC“. Wählen Sie eine geeignete Flugnummer.

```
1 INSERT INTO Flight(FlightID, Orig_IATA, Dest_IATA) VALUES
2 ('LH-205', 'FRA', 'ABC');
```

Data Output Messages Notifications

ERROR: insert or update on table "flight" violates foreign key constraint "fk_flight_flightdes_airport"
DETAIL: Key (dest_iata)=(ABC) is not present in table "airport".
SQL state: 23503

Bild 6, Insert Flight + Fehlermeldung

Flight könnte in diesem Fall nicht hinzugefügt, weil in Tabelle Airport IATA = „ABC“ nicht gibt (Es gibt keinen Flughafen mit dieser IATA) und IATA, in diesem Fall Destination IATA

wird als Fremd Key (referenziert Tabelle Airport) in Tabelle Flight verwendet.

3) Ein Wartungsvorgang mit Flugfreigabe für die Maschine „D-XXXX“.

```
1 Insert into Maintenance(tID, PlaneID, Datum) values
2 (1, 'D-XXXX', '2022-12-11');
```

Data Output Messages Notifications

ERROR: insert or update on table "maintenance" violates foreign key constraint "fk_maintena_planemain_plane"
DETAIL: Key (planeid)=(D-XXXX) is not present in table "plane".
SQL state: 23503

Bild 7, Insert Maintenance + Fehlermeldung

Maintenance könnte in diesem Fall nicht hinzugefügt, weil in Tabelle Plane kein Plane mit PlaneID = „D-XXXX“ gibt und PlaneID ist gleichzeitig auch Fremd Key von Maintenance (referenziert Tabelle Plane).

4) Löschen Sie den Flughafen London Heathrow.

```
1 delete from Airport where IATA = 'LHR';
```

Data Output Messages Notifications

ERROR: update or delete on table "airport" violates foreign key constraint "fk_flight_flightdes_airport" on table "flight"
DETAIL: Key (iata)=(LHR) is still referenced from table "flight".
SQL state: 23503

Bild 8, Delete from Airport + Fehlermeldung

Tabelle Flight verwendet IATA als Fremd Key bzw. referenziert Tabelle Airport durch Verwendung von IATA (DestIATA oder OrigIATA). In diesem Fall wird in dem Bild 8 genannte IATA = „LHR“ in Tabelle Flight als Referenz auf Tabelle Airport verwendet, und deshalb kann diese IATA aus Airport-Tabelle nicht entfernt werden.

4.3

Wenn es um Fremd Key Constraints zwischen Departure und Flight geht, es gibt mindestens zwei Stellen die beachtet werden müssen bzw. verändert werden müssen.

```
alter table Flight
  add constraint FK_FLIGHT_FLIGHTDES_AIRPORT foreign key (Dest_IATA)
  references Airport (IATA)
  on delete cascade on update restrict;
```

Bild 9, Fremd Key Constraint Dest_IATA -> on delete **cascade**

```
alter table Departure
  add constraint FK_DEPARTUR_DEPARTURE_FLIGHT foreign key (FlightID)
  references Flight (FlightID)
  on delete cascade on update restrict;
```

Bild 10, Fremd Key Constraint FlightID -> on delete **cascade**

1. Löschen Sie nun den Flughafen „Chicago O’Hare International Airport“ mit dem IATA Code „ORD“. Dabei sollten keine weiteren Fehlermeldungen ausgegeben werden.

```
1 delete from Airport where IATA = 'ORD';
```

Data Output **Messages** Notifications

DELETE 1

Query returned successfully in 161 msec.

Bild 11, Delete from Airport.. **ORD**

2. Überprüfen Sie, ob weitere Datensätze, evtl. auch in anderen Tabellen, verändert wurden. Falls Sie Änderungen entdecken, protokollieren Sie diese.

Nachdem „Chicago O’Hare International Airport“ gelöscht ist, wurden auch die Änderungen in zwei anderen Tabellen bemerkt und zwar in Tabellen Flight und Departure.

In Tabelle Flight, sind alle Flüge die IATA „ORD“ verwenden, gelöscht worden, und zwar Flug mit FlightID „LH-103“ und Flug mit FlightID „LH-203“.

	flightid [PK] character varying (254) 	dest_iata character varying (254) 	orig_iata character varying (254) 
1	LH-100	ATL	FRA
2	LH-102	DXB	FRA
3	LH-104	HND	FRA
4	LH-105	LHR	FRA
5	LH-106	LAX	FRA
6	LH-107	HKG	FRA
7	LH-108	CDG	FRA
8	LH-109	DFW	FRA
9	LH-110	IST	FRA
10	LH-200	ATL	LHR
11	LH-201	PEK	LHR
12	LH-202	DXB	LHR
13	LH-204	FRA	LHR

Bild 12, Tabelle Flight, nachdem IATA **ORD** aus Airport entfernt wurde

In Tabelle Departure sind alle Flüge mit FlightID „103“ bzw. FlightID „104“ gelöscht worden, weil diese Flüge aus Tabelle Flight entfernt sind.




	flightid [PK] character varying (254) 	datum [PK] date 	planeid character varying (254) 
1	LH-100	2018-01-10	D-ABBL
2	LH-100	2018-02-10	D-ABBL
3	LH-100	2018-03-10	D-ABBL
4	LH-100	2018-04-10	D-ABBL
5	LH-100	2018-05-10	D-ABBL
6	LH-100	2018-06-10	D-ABBL
7	LH-100	2018-07-10	D-ABBK
8	LH-100	2018-08-10	D-ABBK
9	LH-102	2018-02-10	D-ABBB
10	LH-102	2018-04-10	D-ABBB
11	LH-102	2018-06-10	D-ABBB
12	LH-104	2018-03-10	D-ABBB
13	LH-104	2018-05-10	D-ABBB
14	LH-104	2018-07-10	D-ABBB
15	LH-104	2018-09-10	D-ABBB
16	LH-105	2018-01-10	D-ABBB
17	LH-105	2018-02-10	D-ABBB
18	LH-105	2018-03-10	D-ABBB
19	LH-200	2018-01-10	D-ABCA
20	LH-200	2018-02-10	D-ABCA
21	LH-201	2018-03-10	D-ABCA
22	LH-201	2018-05-10	D-ABBR

Bild 13, Tabelle Departure, nachdem IATA **ORD** aus Airport entfernt wurde (1)

23	LH-202	2018-05-10	D-ABCA
24	LH-202	2018-09-10	D-ABCA
25	LH-204	2018-01-10	D-ABBF
26	LH-204	2018-03-10	D-ABBI
27	LH-204	2018-08-10	D-ABBM

Bild 13, Tabelle Departure, nachdem IATA **ORD** aus Airport entfernt wurde (2)

3. Interpretieren Sie, welche Auswirkungen on delete cascade auf Ihre Daten hat.

Wie wir schon in 4.2 genannt haben, wenn das Constraint zu „on delete cascade“ ändert, werden alle Daten die davon Abhängig sind gelöscht werden. Es ist aber nicht nur auf „delete“ beschränkt, wenn das Schlüsselwort „cascade“ zu einem Query hinzugefügt wird, dann werden bei der Ausführung eines Querys beide Child und Parent Tabelle geändert (z.B bei update oder drop).

Probieren Sie nun, den Flughafen Frankfurt (IATA-Code „FRA“) zu löschen. Passiert dabei das Gleiche wie beim vorherigen Flughafen? Wenn nein, was unterscheidet den Flughafen FRA von ORD?

```
1 delete from Airport where IATA = 'FRA';
```

Data Output Messages Notifications

ERROR: update or delete on table "airport" violates foreign key constraint "fk_flight_flightori_airport" on table "flight"
DETAIL: Key (iata)=(FRA) is still referenced from table "flight".
SQL state: 23503

Bild 14, Delete from Airport + Fehlermeldung

Flughafen Frankfurt könnte nicht entfernt werden weil, die Änderungen die vorgenommen worden sind um die Entfernung von Flughafen Chicago möglich zu machen, nicht genug sind die Flughafen Frankfurt zu entfernen. Das ist der Fall, weil, Flughafen Chicago nur als Destination verwendet wird, und Flughafen Frankfurt als Destination sowohl als auch Origin verwendet ist. Um Entfernung Flughafens Frankfurt möglich zu machen, muss in Tabelle Flight die Constraint mit Orig_IATA als Fremd Key auch von „on delete **restrict** on update restrict“ auf „on delete **cascade** on update restrict“ geändert werden (siehe Bild 15).

```
alter table Flight  
add constraint FK_FLIGHT_FLIGHTORI_AIRPORT foreign key (Orig_IATA)  
references Airport (IATA)  
on delete restrict on update restrict;
```

Bild 15, Stelle wo eine Änderung vorgenommen werden muss