SCENARIO

As a travel company we are having intercity expeditions with our own buses. We carry individual passengers as well as the customers of travel agencies. We give discounts to students and owners of our special cards. Travel agencies pay for their customers all together. In our expeditions, we do not allow taking or dropping passengers after expedition started. We provide breakfast on some of our expeditions. We take vehicles which are more than 5-years-old off from services. We always do get maintenance of our vehicles 1 month before its due date. There are different brands and model buses. All the buses are in 2+1 seat arrangement. In all expeditions there must be an assistant alongside driver. In long-distance travelling two drivers must be assigned. We must know that drivers have driver license of D level and phone number besides their general information. If a driver lives in a destination city, he may be preferred for that expedition. Assistants take different level training courses and according to these training courses they are assigned to each expedition. We want to know the time, success level and type of training taken by each assistant.

Seyahat firması olarak kendi araçlarımız ile şehirlerarası seferler düzenlemekteyiz. Bireysel müşterilere ve turizm acentelerinin müşterilerine hizmet vermekteyiz. Bireysel müşterilerde öğrenci veya kart sahiplerine indirim sağlamaktayız. Seyahat acentelerinin müşterilerinin ücretleri acenteden toplu olarak alınmaktadır. Seferlerimizde ara yerlerde yolcu alınıp indirilmemektedir. Bazı seferlerimizde kahvaltı verilmektedir. 5 yılını tamamlayan araçları hizmetten kaldırıyoruz. Araçlarımızın bakım tarihleri gelmeden 1 ay önce mutlaka bakımını yaptırırız. Farklı marka ve modellerde araçlar bulunabilmektedir. Araçların tamamı 2+1 koltuk düzenindedir. Her seyahatte şoförün yanında mutlaka bir muavin bulunması gerekiyor. Uzun seferlerde iki şoför bulundurma zorunluluğu vardır. Şoförlerin temel bilgilerine ilave olarak D sınıfı ehliyeti olduğunu ve cep telefonunu bilmek istiyoruz. Şoförler gidilen şehirde yaşıyorsa öncelik onlara verilmektedir. Muavinlerin aldıkları eğitimlere göre gidecekleri seferler belirlenmektedir. Eğitimlerin ne zaman alındığı, eğitimdeki başarısını ve eğitimin tipini bilmek istiyoruz.

ENTITIES

Entity-1	Entity-2	Entity-3	Entity-4	Entity-5
CITY	DRIVER	ASSISTANT	TRAINING	PLAN_EXPEDITION
Entity-6	Entity-7	Entity-8	Entity-9	Entity-10
REAL_EXPEDITION	MAINTENANCE	VEHICLE	PASSENGER	AGENCY
Entity-11	Entity-12			
TICKET	PAYMENT			

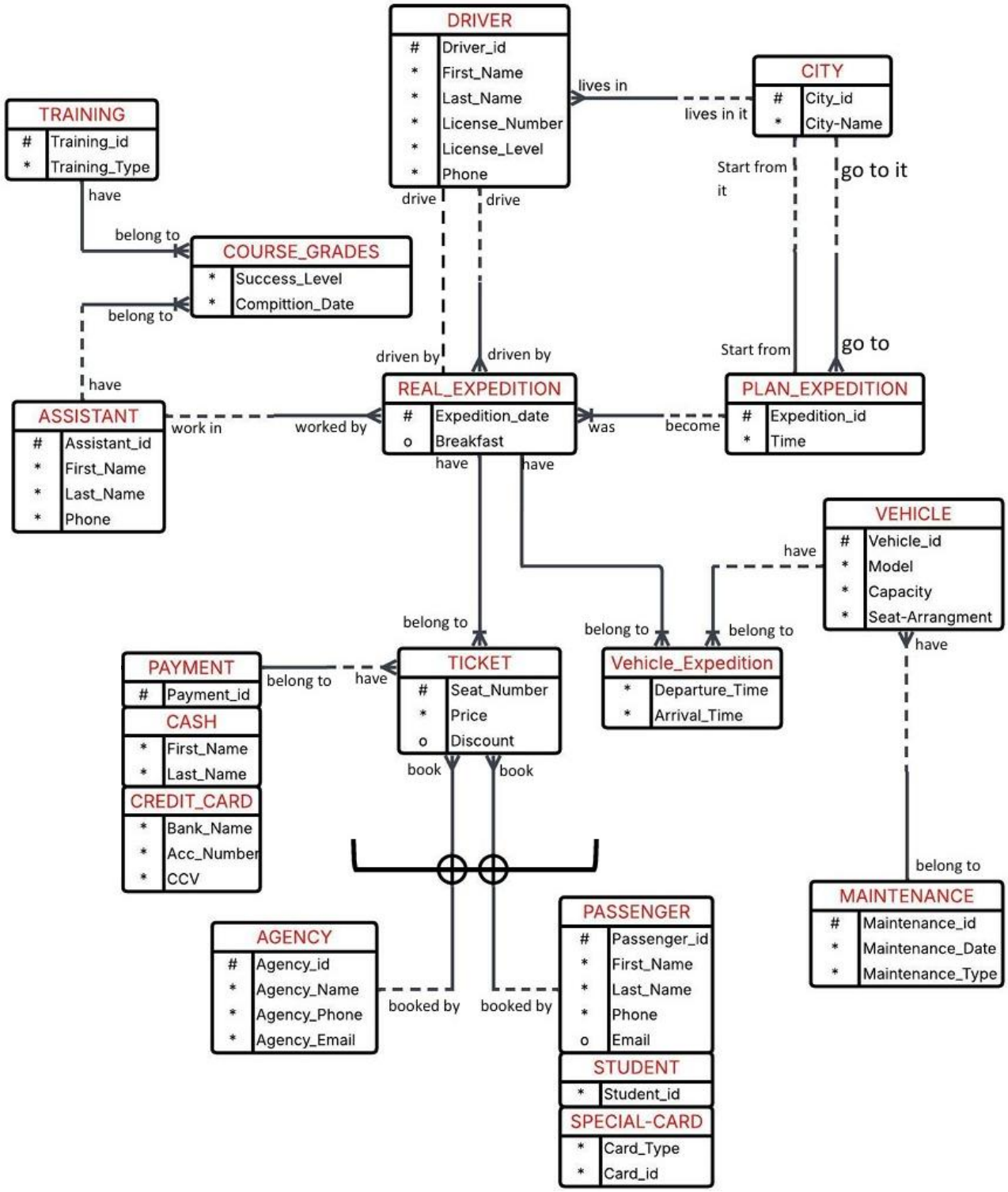
ATTRIBUTES

CITY	DRIVER	ASSISTANT	TRAINING	PLAN_EXPEDITION
# City_id	# Driver_id	# Assistant_id	# Training_id	# Expedition_id
* City_Name	* First_Name	* First_Name	* Training_Type	* Time
	* Last_Name	* Last_Name		
	* License_Number	* Phone		
	* License_Level			
	* Phone			
REAL_EXPEDITION	MAINTENANCE	VEHICLE	PASSENGER	AGENCY
# Expedition_Date	# Maintenance_id	# Vehicle_id	# Passenger_id	# Agency_id
o Breakfast	* Maintenance_Date	* Model	* First_Name	* Agency_Name
	* Maintenance_Type	* Capacity	* Last_Name	* Agency_Phone
		* Seat_Arrangement	* Phone	* Agency_Email
			o Email	
			STUDENT	
			* Student_id	
			SPECIAL_CARD	
			* Card_Type	
			* Card_id	
TICKET	PAYMENT			
# Seat_Number	# Payment_id			
* Price	CASH			
o Discount	* First_Name			
	* Last_Name			
	CREDIT_CARD			
	* Bank_Name			
	* Acc_Nubmer			
	* CVV			

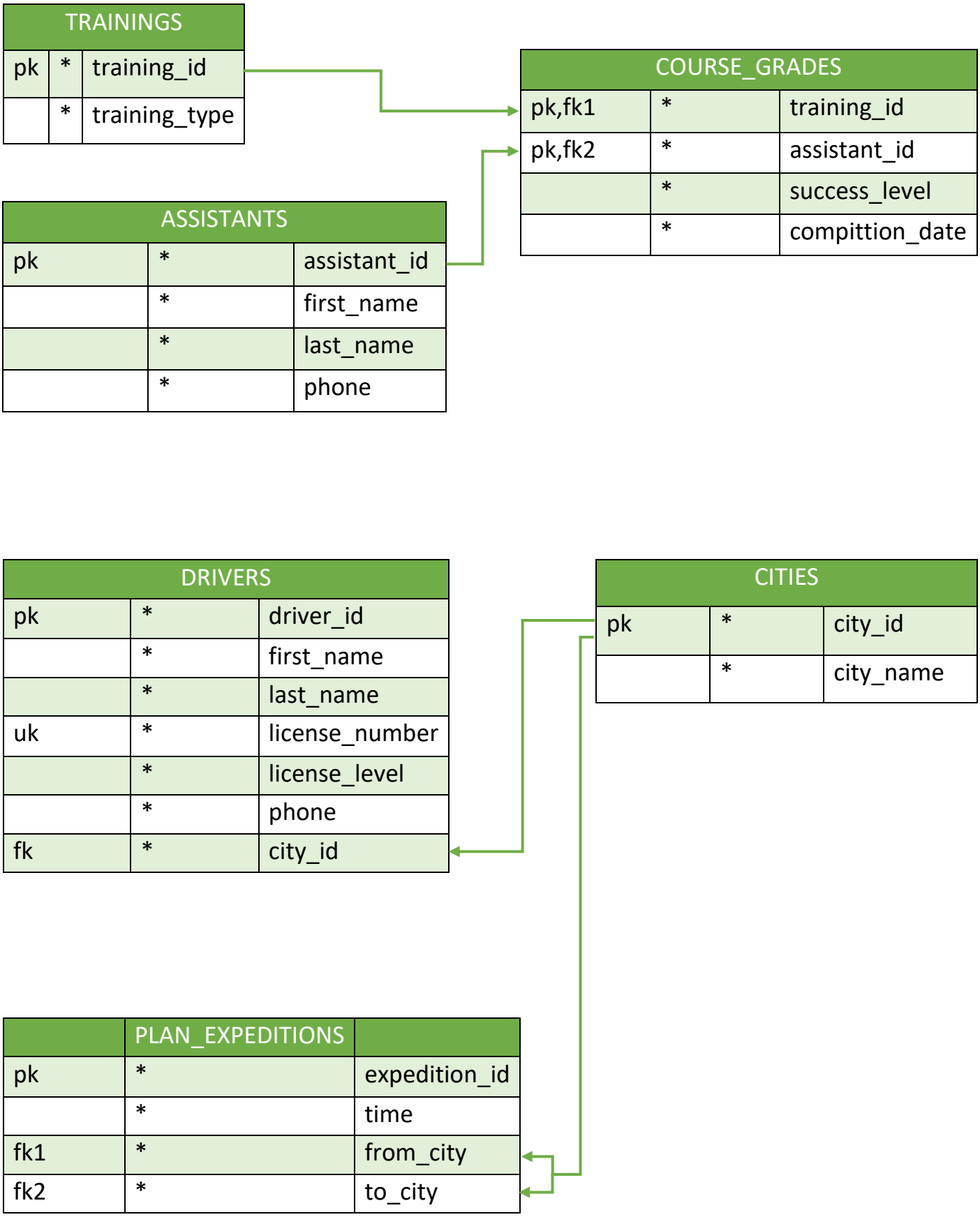
MATRIX DIAGRAM

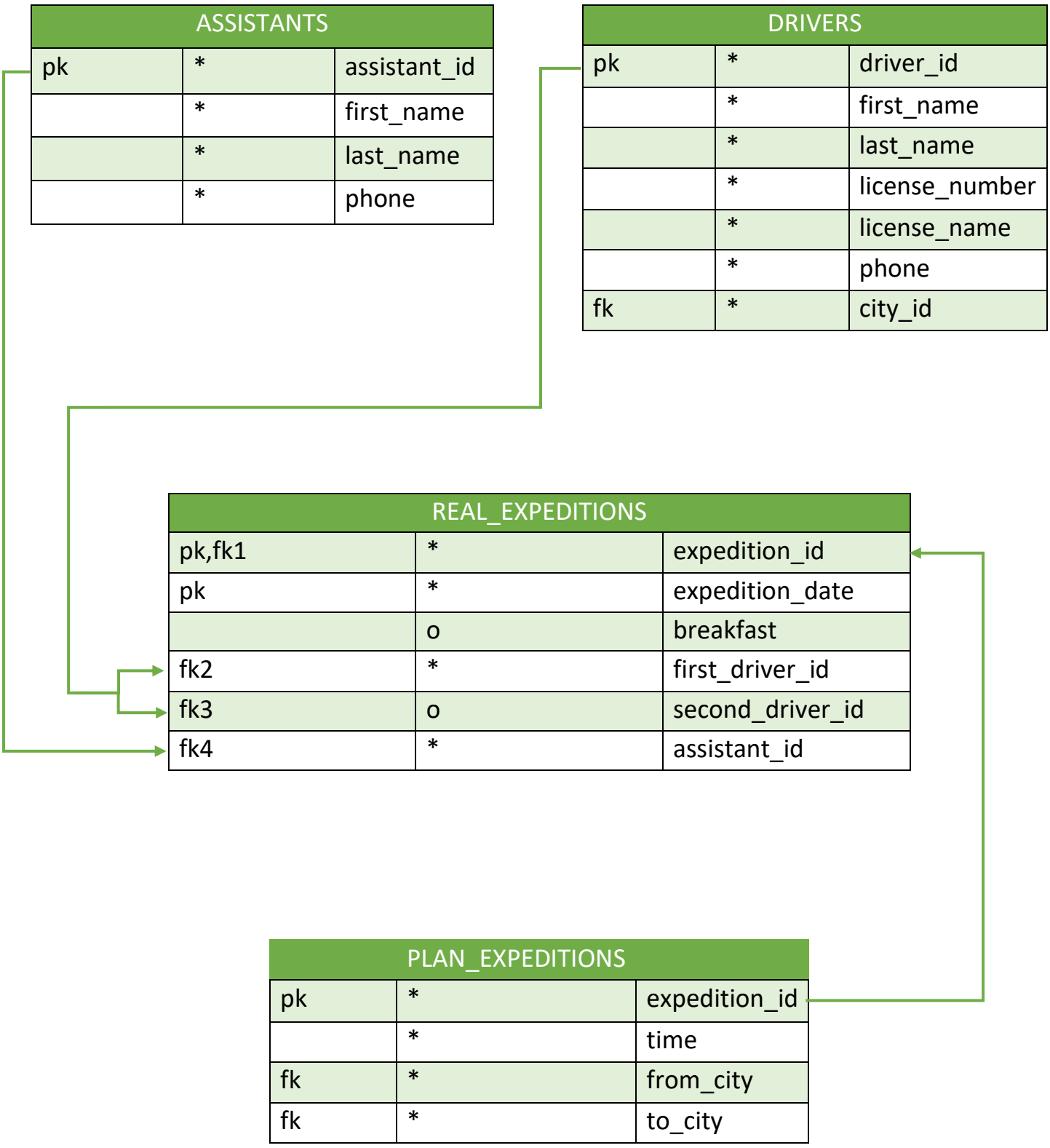
	CITY	DRIVER	ASSISTANT	TRAINING	PLAN_ EXPEDITION	REAL_ EXPEDITION	MAINTENANCE	VEHICLE	PASSENGER	AGENCY	TICKET	PAYMENT
CITY		lives in it			start from, go to							
DRIVER	lives in					drive, drive		drive				
ASSISTANT				take		work in						
TRAINING			taken by									
PLAN_ EXPEDITION	start from, go to					become						
REAL_ EXPEDITION		driven by, driven by	worked by		was			have			have	
MAINTENANCE								belong to				
VEHICLE		driven by				belong to	have					
PASSENGER											book	
AGENCY											book	
TICKET						belong to			booked by	booked by		have
PAYMENT											belong to	

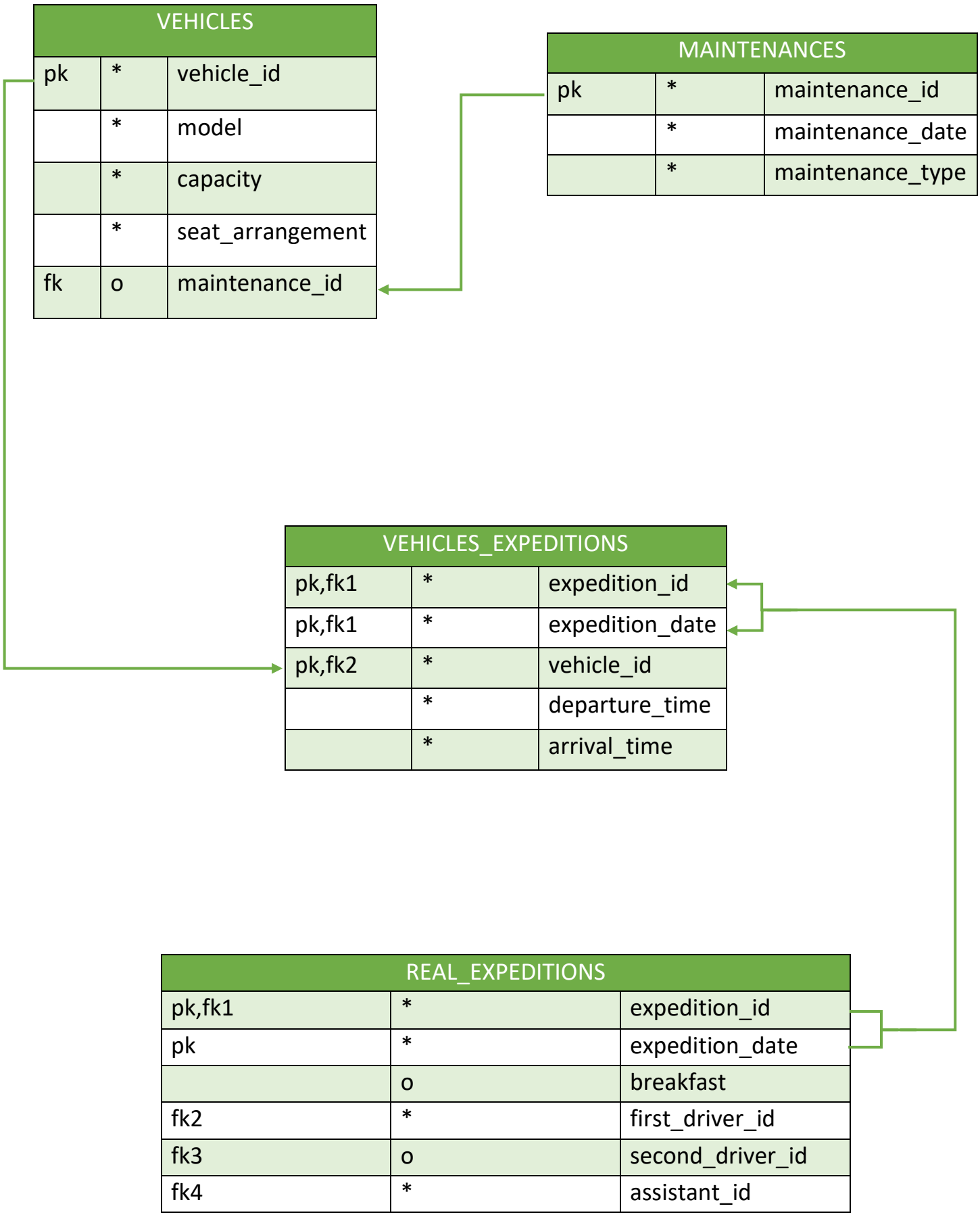
ERD

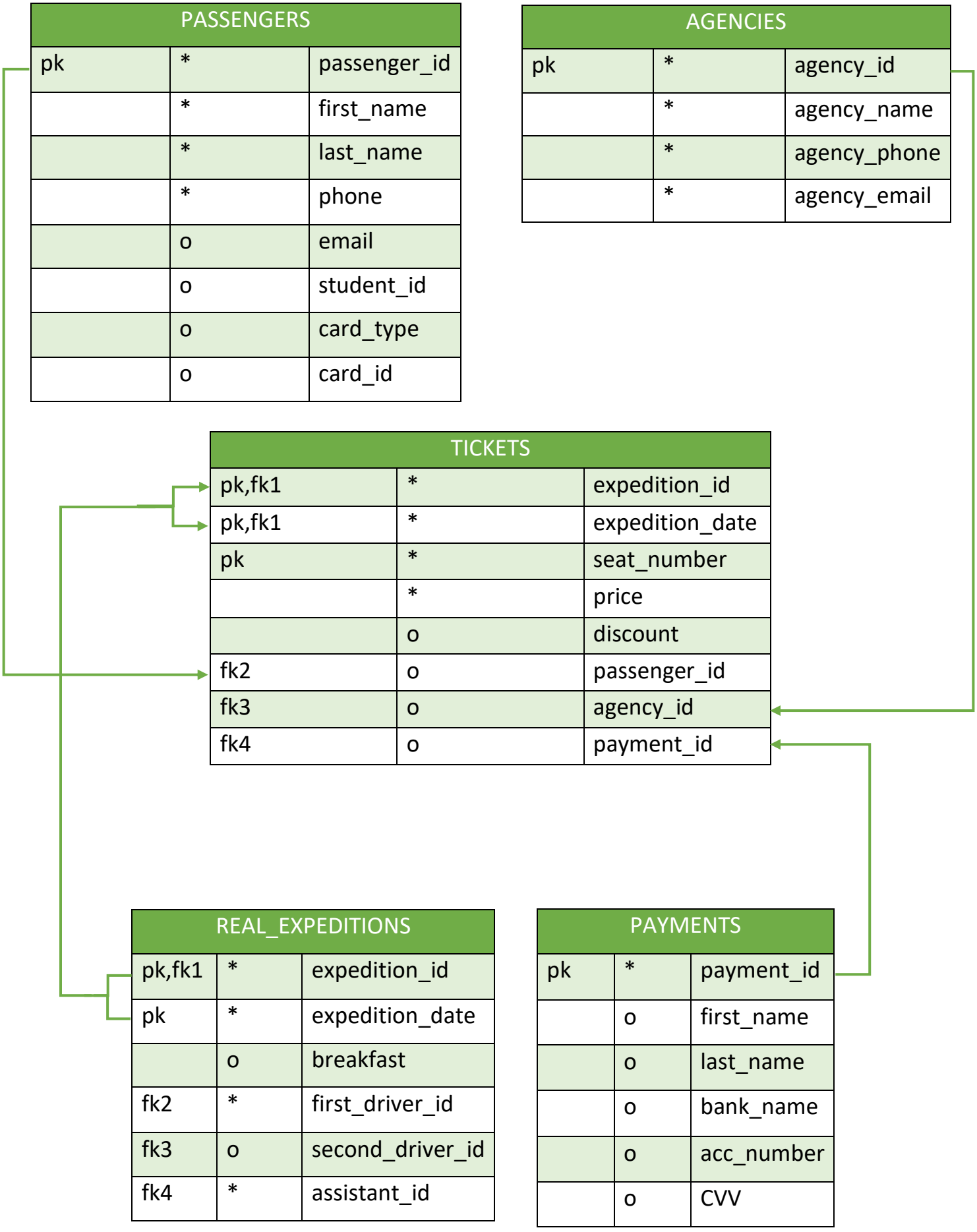

DRIVER
Driver_id
* First_Name
* Last_Name
* License_Number
* License_Level
* Phone
lives in
lives in it
CITY
City_id
* City-Name
TRAINING
Training_id
* Training_Type
have
belong to
COURSE_GRADES
* Success_Level
* Compittion_Date
belong to
drive
drive
Start from it
go to it
driven by
driven by
Start from
go to
have
ASSISTANT
Assistant_id
* First_Name
* Last_Name
* Phone
work in
worked by
REAL_EXPEDITION
Expedition_date
o Breakfast
was
become
PLAN_EXPEDITION
Expedition_id
* Time
have
have
VEHICLE
Vehicle_id
* Model
* Capacity
* Seat-Arrangment
have
belong to
belong to
belong to
have
PAYMENT
Payment_id
CASH
* First_Name
* Last_Name
CREDIT_CARD
* Bank_Name
* Acc_Number
* CCV
belong to
have
TICKET
Seat_Number
* Price
o Discount
Vehicle_Expedition
* Departure_Time
* Arrival_Time
book
book
belong to
MAINTENANCE
Maintenance_id
* Maintenance_Date
* Maintenance_Type
AGENCY
Agency_id
* Agency_Name
* Agency_Phone
* Agency_Email
booked by
booked by
PASSENGER
Passenger_id
* First_Name
* Last_Name
* Phone
o Email
STUDENT
* Student_id
SPECIAL-CARD
* Card_Type
* Card_id

MAPING


TRAININGS
pk * training_id
* training_type
COURSE_GRADES
pk,fk1 * training_id
pk,fk2 * assistant_id
* success_level
* compittion_date
ASSISTANTS
pk * assistant_id
* first_name
* last_name
* phone
DRIVERS
pk * driver_id
* first_name
* last_name
uk * license_number
* license_level
* phone
fk * city_id
CITIES
pk * city_id
* city_name
PLAN_EXPEDITIONS
pk * expedition_id
* time
fk1 * from_city
fk2 * to_city


ASSISTANTS
pk * assistant_id
* first_name
* last_name
* phone
DRIVERS
pk * driver_id
* first_name
* last_name
* license_number
* license_name
* phone
fk * city_id
REAL_EXPEDITIONS
pk,fk1 * expedition_id
pk * expedition_date
o breakfast
fk2 * first_driver_id
fk3 o second_driver_id
fk4 * assistant_id
PLAN_EXPEDITIONS
pk * expedition_id
* time
fk * from_city
fk * to_city


VEHICLES
pk * vehicle_id
* model
* capacity
* seat_arrangement
fk o maintenance_id
MAINTENANCES
pk * maintenance_id
* maintenance_date
* maintenance_type
VEHICLES_EXPEDITIONS
pk,fk1 * expedition_id
pk,fk1 * expedition_date
pk,fk2 * vehicle_id
* departure_time
* arrival_time
REAL_EXPEDITIONS
pk,fk1 * expedition_id
pk * expedition_date
o breakfast
fk2 * first_driver_id
fk3 o second_driver_id
fk4 * assistant_id


PASSENGERS
pk * passenger_id
* first_name
* last_name
* phone
o email
o student_id
o card_type
o card_id
AGENCIES
pk * agency_id
* agency_name
* agency_phone
* agency_email
TICKETS
pk,fk1 * expedition_id
pk,fk1 * expedition_date
pk * seat_number
* price
o discount
fk2 o passenger_id
fk3 o agency_id
fk4 o payment_id
REAL_EXPEDITIONS
pk,fk1 * expedition_id
pk * expedition_date
o breakfast
fk2 * first_driver_id
fk3 o second_driver_id
fk4 * assistant_id
PAYMENTS
pk * payment_id
o first_name
o last_name
o bank_name
o acc_number
o CVV

How to create a table with SQL?

Language SQL Rows 10 Clear Command Find Tables Save Run

```sql
CREATE TABLE TICKETS (
    EXPEDITION_ID   NUMBER(10,0),
    EXPEDITION_DATE DATE,
    SEAT_NUMBER VARCHAR(25),
    PRICE   NUMBER(5,0) NOT NULL,
    DISCOUNT    NUMBER(10,0),
    AGENCY_ID   NUMBER(10,0),
    PASSENGER_ID    NUMBER(10,0),
    PAYMENT_ID  NUMBER(10,0),

    CONSTRAINT pk_TICKETS PRIMARY KEY (EXPEDITION_ID, EXPEDITION_DATE, SEAT_NUMBER),
    CONSTRAINT fk_TICKETS_AGENCIES FOREIGN KEY (AGENCY_ID) REFERENCES AGENCIES(AGENCY_ID),
    CONSTRAINT fk_TICKETS_PASSENGERS FOREIGN KEY (PASSENGER_ID) REFERENCES PASSENGERS(PASSENGER_ID),
    CONSTRAINT fk_TICKETS_PAYMENTS FOREIGN KEY (PAYMENT_ID) REFERENCES PAYMENTS(PAYMENT_ID)
);
```

How to insert data in the table with SQL?

Language SQL Rows 10 Clear Command Find Tables Save Run

```sql
INSERT INTO TICKETS (expedition_id, expedition_date, seat_number, price, discount, agency_id, passenger_id, payment_id) VALUES
(1, TO_DATE('2025-05-05', 'YYYY-MM-DD'), 'A1', 50, 0.025, NULL, 1, 500);

INSERT INTO TICKETS (expedition_id, expedition_date, seat_number, price, discount, agency_id, passenger_id, payment_id) VALUES
(1, TO_DATE('2025-05-05', 'YYYY-MM-DD'), 'B2', 50, 0.025, NULL, 2, 505);

INSERT INTO TICKETS (expedition_id, expedition_date, seat_number, price, discount, agency_id, passenger_id, payment_id) VALUES
(1, TO_DATE('2025-05-05', 'YYYY-MM-DD'), 'D1', 50, 0.025, NULL, 4, 510);

INSERT INTO TICKETS (expedition_id, expedition_date, seat_number, price, discount, agency_id, passenger_id, payment_id) VALUES
(1, TO_DATE('2025-05-05', 'YYYY-MM-DD'), 'E2', 50, 0.025, NULL, 5, 515);

INSERT INTO TICKETS (expedition_id, expedition_date, seat_number, price, discount, agency_id, passenger_id, payment_id) VALUES
(2, TO_DATE('2025-05-12', 'YYYY-MM-DD'), 'F3', 60, NULL, 1, NULL, 520);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.

0.00 seconds
```

tr_a832_sql_s58 tr_a832_sql_s58 en Oracle APEX 22.2.1

How to display the whole data from the table with SQL?

Language SQL Rows 10 Clear Command Find Tables Save Run

```sql
SELECT *
FROM TICKETS;
```

Results Explain Describe Saved SQL History

EXPEDITION_ID	EXPEDITION_DATE	SEAT_NUMBER	PRICE	DISCOUNT	AGENCY_ID	PASSENGER_ID	PAYMENT_ID
2	12-May-2025	H5	60	-	3	-	530
2	12-May-2025	I6	60	-	4	-	535
1	05-May-2025	A1	50	0	-	1	500
2	12-May-2025	G4	60	-	2	-	525
2	12-May-2025	F3	60	-	1	-	520
1	05-May-2025	E2	50	0	-	5	515
1	05-May-2025	B2	50	0	-	2	505
1	05-May-2025	D1	50	0	-	4	510
2	12-May-2025	J7	60	-	5	-	540

9 rows returned in 0.00 seconds Download

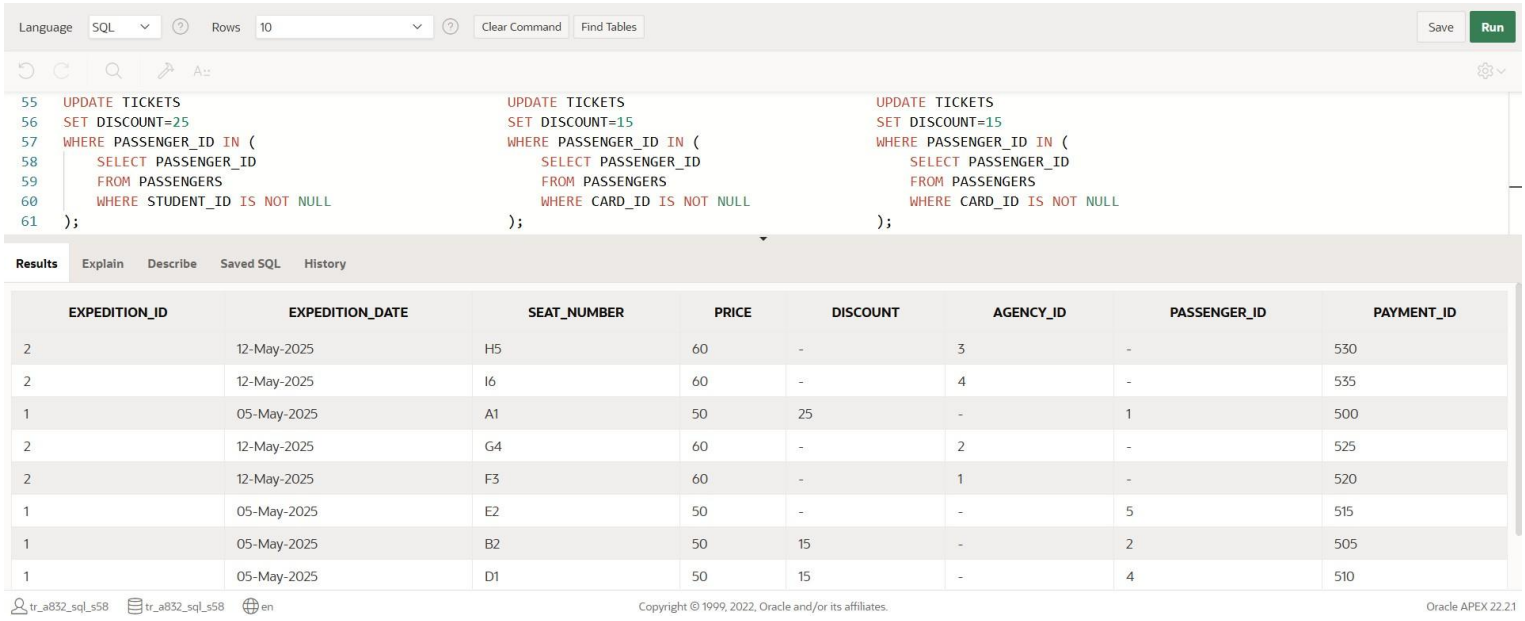
tr_a832_sql_s58 tr_a832_sql_s58 en Oracle APEX 22.2.1

How to update the data in the table with SQL?

EXPEDITION_ID	EXPEDITION_DATE	SEAT_NUMBER	PRICE	DISCOUNT	AGENCY_ID	PASSENGER_ID	PAYMENT_ID
2	12-May-2025	H5	60	-	3	-	530
2	12-May-2025	I6	60	-	4	-	535
1	05-May-2025	A1	50	0	-	1	500
2	12-May-2025	G4	60	-	2	-	525
2	12-May-2025	F3	60	-	1	-	520
1	05-May-2025	E2	50	0	-	5	515
1	05-May-2025	B2	50	0	-	2	505
1	05-May-2025	D1	50	0	-	4	510
2	12-May-2025	J7	60	-	5	-	540

9 rows returned in 0.00 seconds Download

Here for example, we noticed that the discount values are 0 for the whole passengers even if they are students or having a special card so we need to update it.



The first one we updated the DISCOUNT for the students like passenger with id 1, the second one for who has a special card like passengers with id 2,4 and the third one for who neither a student nor has a special card like passenger with id 5.

Note: The student who has student_id 3 doesn't have a ticket.

Here some examples about Subquery:

1)

```
SELECT VEHICLE_ID, MODEL, CAPACITY
FROM VEHICLES
WHERE CAPACITY IN (
    SELECT CAPACITY
    FROM VEHICLES
    WHERE CAPACITY > 30
)
```

Results | Explain | Describe | Saved SQL | History

VEHICLE_ID	MODEL	CAPACITY
1010	01-Jan-2024	35
1040	01-Jan-2024	35
1000	01-Jan-2025	45
1020	01-Jan-2025	40

4 rows returned in 0.00 seconds Download

2)

```
SELECT *
FROM REAL_EXPEDITIONS
WHERE BREAKFAST IN (
    SELECT BREAKFAST
    FROM REAL_EXPEDITIONS
    WHERE EXPEDITION_ID > 8
)
```

Results | Explain | Describe | Saved SQL | History

EXPEDITION_ID	FIRST_DRIVER_ID	ASSISTANT_ID	EXPEDITION_DATE	BREAKFAST	SECOND_DRIVER_ID
4	103	13	09-May-2025	Cheese	-
9	106	18	15-May-2025	Cheese	-
3	107	12	10-May-2025	Pizza	-
10	102	19	11-May-2025	Pizza	-

4 rows returned in 0.02 seconds Download

tr_a832_sql_s58 tr_a832_sql_s58 en

Oracle APEX 22.2.1

3)

```
SELECT *
FROM TICKETS
WHERE PASSENGER_ID IN (
    SELECT PASSENGER_ID
    FROM PASSENGERS
    WHERE CARD_ID IS NOT NULL
)
```

Results | Explain | Describe | Saved SQL | History

EXPEDITION_ID	EXPEDITION_DATE	SEAT_NUMBER	PRICE	DISCOUNT	AGENCY_ID	PASSENGER_ID	PAYMENT_ID
1	05-May-2025	B2	50	15	-	2	505
1	05-May-2025	D1	50	15	-	4	510

2 rows returned in 0.02 seconds Download

tr_a832_sql_s58 tr_a832_sql_s58 en

Oracle APEX 22.2.1

Some examples about JOIN:

1)

```
SELECT ASSISTANT_ID, FIRST_NAME, LAST_NAME, SUCCESS_LEVEL
FROM ASSISTANTS NATURAL JOIN COURSE_GRADES;
```

ASSISTANT_ID	FIRST_NAME	LAST_NAME	SUCCESS_LEVEL
14	Yasin	Fahmi	AA
15	Layla	Ibrahim	CC
16	Khalid	Salem	BA
10	Ali	Hassan	AA
17	Noura	Mahmoud	BB
12	Omar	Yousef	BA
13	Mona	Adel	DC
18	Fadi	Saad	DC
19	Rania	Nasr	AA
11	Sara	Kamel	BB

2)

```
SELECT DRIVER_ID, FIRST_NAME, LAST_NAME, CITY_NAME
FROM DRIVERS D JOIN CITIES C
ON (D.CITY_ID=C.CITY_ID)
```

DRIVER_ID	FIRST_NAME	LAST_NAME	CITY_NAME
101	Mehmet	Demir	Trabzon
108	Can	Doğan	Istanbul
102	Ayşe	Kaya	Adana
104	Emre	Şahin	Izmir
106	Ali	Öztürk	Antalya
109	Hüseyin	Güneş	Bursa
105	Zeynep	Koç	Ankara
107	Elif	Arslan	Eskisehir
100	Ahmet	Yılmaz	Gaziantep

3)

```
SELECT VEHICLE_ID, MODEL, CAPACITY, MAINTENANCE_TYPE
FROM VEHICLES V LEFT OUTER JOIN MAINTENANCES M
ON (V.MAINTENANCE_ID=M.MAINTENANCE_ID);
```

VEHICLE_ID	MODEL	CAPACITY	MAINTENANCE_TYPE
1010	01-Jan-2024	35	-
1030	01-Jan-2023	25	-
1000	01-Jan-2025	45	-
1040	01-Jan-2024	35	-
1020	01-Jan-2025	40	-
1050	01-Jan-2020	30	Full Service

6 rows returned in 0.01 seconds Download

Some examples about GROUP BY:

1)

```
SELECT EXPEDITION_ID, SUM(PRICE)
FROM TICKETS
GROUP BY EXPEDITION_ID;
```

EXPEDITION_ID	SUM(PRICE)
1	200
2	300

2 rows returned in 0.01 seconds

2)

```
SELECT EXPEDITION_DATE, SUM(PRICE)
FROM TICKETS
GROUP BY EXPEDITION_DATE
HAVING COUNT(EXPEDITION_ID)=5;
```

EXPEDITION_DATE	SUM(PRICE)
12-May-2025	300

1 rows returned in 0.01 seconds

3)

```
SELECT EXPEDITION_ID, SUM(DISCOUNT)
FROM TICKETS
GROUP BY EXPEDITION_ID
HAVING SUM(DISCOUNT) IS NOT NULL;
```

EXPEDITION_ID	SUM(DISCOUNT)
1	55

1 rows returned in 0.00 seconds

```
SELECT EXPEDITION_ID, SUM(DISCOUNT)
FROM TICKETS
GROUP BY EXPEDITION_ID;
```

EXPEDITION_ID	SUM(DISCOUNT)
1	55
2	-

2 rows returned in 0.01 seconds

Some examples about DATA FUNCTIONS:

1)

```
SELECT VEHICLE_ID, MODEL, SYSDATE, ROUND(MONTHS_BETWEEN(SYSDATE,MODEL)/12) AS "Years of service"
FROM VEHICLES;
```

Results | Explain | Describe | Saved SQL | History

VEHICLE_ID	MODEL	SYSDATE	Years of service
1010	01-Jan-2024	05-May-2025	1
1030	01-Jan-2023	05-May-2025	2
1000	01-Jan-2025	05-May-2025	0
1040	01-Jan-2024	05-May-2025	1
1020	01-Jan-2025	05-May-2025	0
1050	01-Jan-2020	05-May-2025	5

6 rows returned in 0.01 seconds Download

2)

Language SQL Rows 10 Clear Command Find Tables Save Run

```
SELECT 'Vehicle with id: ' || VEHICLE_ID || ' will be taken off from the service in: ' || ADD_MONTHS(MODEL,12*5) AS "The service details"
FROM VEHICLES;
```

Results | Explain | Describe | Saved SQL | History

The service details
Vehicle with id: 1010 will be taken off from the service in: 01-Jan-2029
Vehicle with id: 1030 will be taken off from the service in: 01-Jan-2028
Vehicle with id: 1000 will be taken off from the service in: 01-Jan-2030
Vehicle with id: 1040 will be taken off from the service in: 01-Jan-2029
Vehicle with id: 1020 will be taken off from the service in: 01-Jan-2030
Vehicle with id: 1050 will be taken off from the service in: 01-Jan-2025

6 rows returned in 0.00 seconds Download

3)

Language SQL Rows 10 Clear Command Find Tables Save Run

```
SELECT 'Vehicle with id: ' || VEHICLE_ID || ' should have maintenance in: ' || LAST_DAY(ADD_MONTHS(MODEL,(12*5)-2)) AS "The maintenance details"
FROM VEHICLES;
```

Results | Explain | Describe | Saved SQL | History

The maintenance details
Vehicle with id: 1010 should have maintenance in: 30-Nov-2028
Vehicle with id: 1030 should have maintenance in: 30-Nov-2027
Vehicle with id: 1000 should have maintenance in: 30-Nov-2029
Vehicle with id: 1040 should have maintenance in: 30-Nov-2028
Vehicle with id: 1020 should have maintenance in: 30-Nov-2029
Vehicle with id: 1050 should have maintenance in: 30-Nov-2024

6 rows returned in 0.01 seconds Download

Some examples about CHARACTER FUNCTIONS:

1)

```
SELECT DECODE(INSTR(EMAIL,'@'),NULL,'Its not an email','Its an email') AS "Checking the email"
FROM PASSENGERS;
```

Checking the email
Its an email
Its an email
Its an email
Its an email
Its an email

5 rows returned in 0.01 seconds Download

2)

```
SELECT DRIVER_ID, FIRST_NAME, SUBSTR(PHONE,3) AS "Drivers phone number without prefix"
FROM DRIVERS;
```

DRIVER_ID	FIRST_NAME	Drivers phone number without prefix
101	Mehmet	05301112233
108	Can	05391122334
102	Ayşe	05550998877
104	Emre	05370112233
106	Ali	05301234567
109	Hüseyin	05321112233
105	Zeynep	05461122334
107	Elif	05330987654
100	Ahmet	05321234567
103	Fatma	05411223344

3)

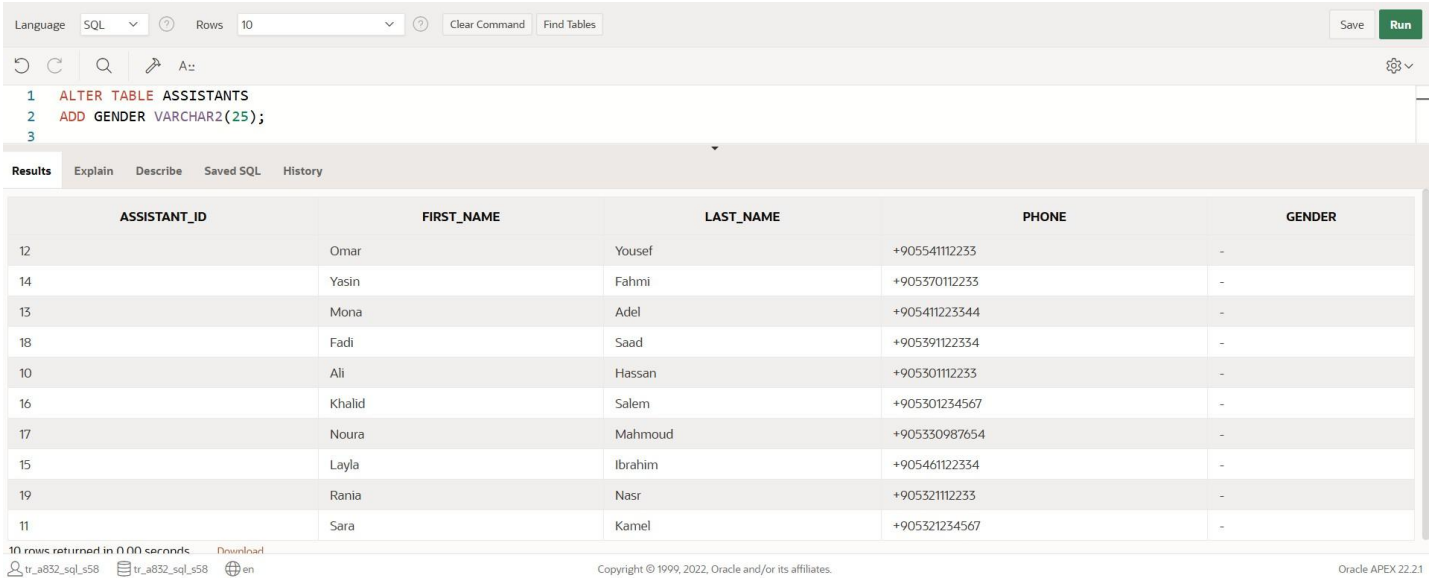
```
SELECT EXPEDITION_ID, PASSENGER_ID, LPAD(SEAT_NUMBER, LENGTH(SEAT_NUMBER)+DISCOUNT, '-')
FROM TICKETS
WHERE DISCOUNT IS NOT NULL;
```

EXPEDITION_ID	PASSENGER_ID	LPAD(SEAT_NUMBER,LENGTH(SEAT_NUMBER)+DISCOUNT,'-')
1	1	----------A1
1	2	----------B2
1	4	----------D1

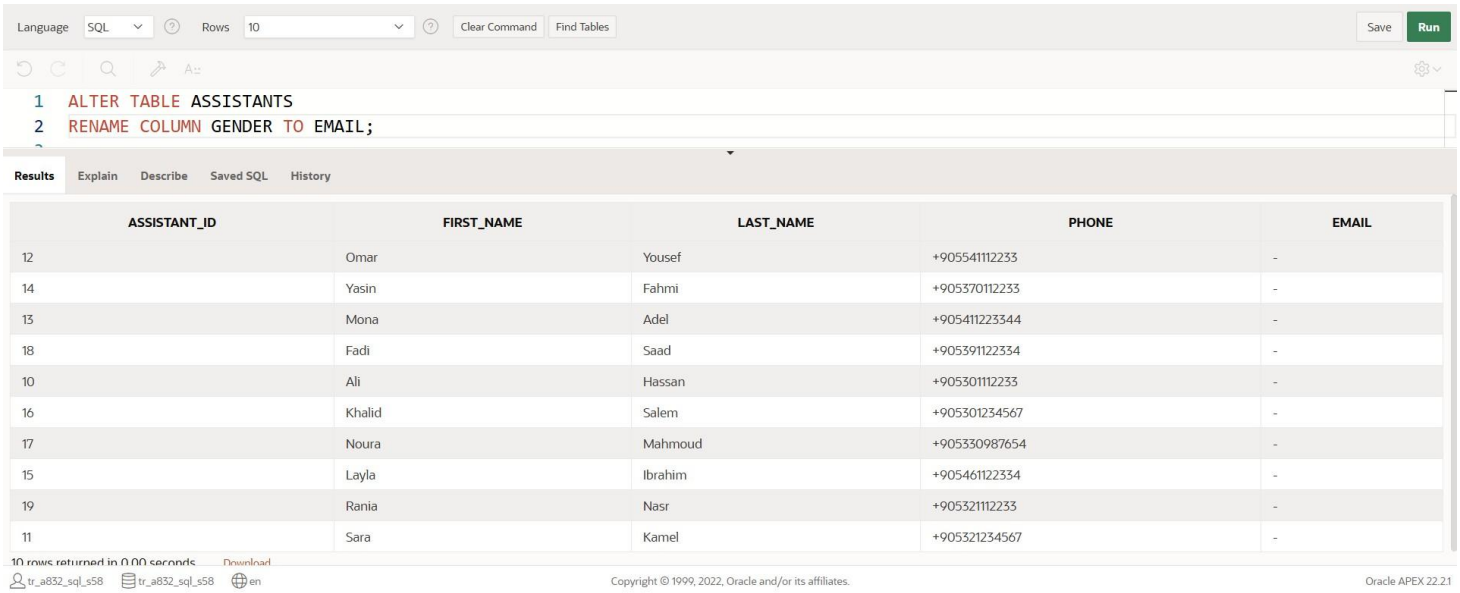
3 rows returned in 0.02 seconds Download

Some examples about ALTER TABLE:

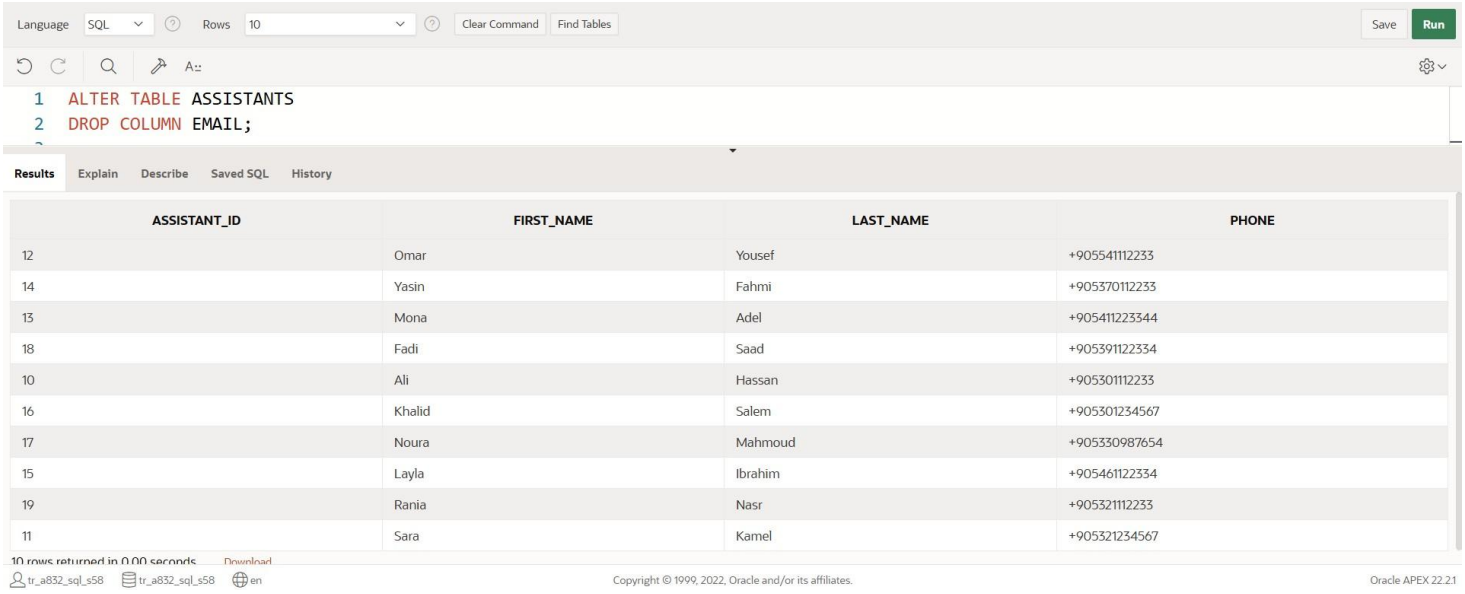
1) How to insert a new column to the table that we have been created it if we noticed that its missing?



2) Now if we noticed that the name is unsuitable how to rename it?



3) Now if the whole column is unsuitable how to delete it?



This project done by:

ANAS A A KHAMAYSA

2210213581

100% English

ABDALHAMID ALBEIK

2210213579

100% English

ABDALJAWWAD TARADEH

2210213586

100% English

LUTFI SHABAN

2110205514

30% English