2018/19

CI5320 Database Application Development

Coursework 1 – Database Design & Development

Name: Leandro Maglianella

K Number: 1833503

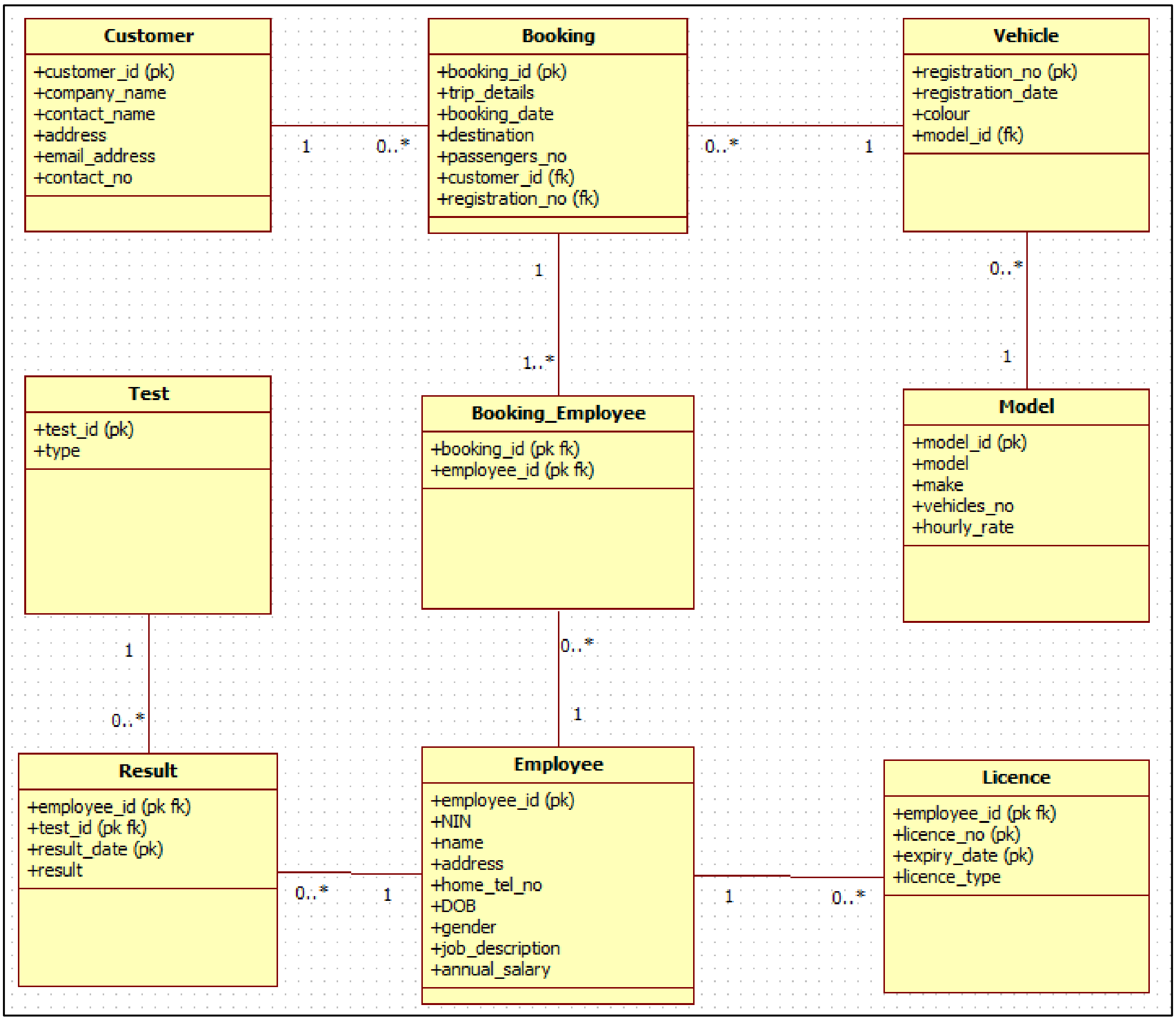
Please save your file as a PDF document

and name it K999999.pdf

(your k number)

1. I declare that the attached work is all my own, and that where I have quoted from, used or referred to the opinions, work or writings of others, these have been fully and clearly acknowledged.
2. I am aware of the consequences of late submission.

Task 1: Entity Relationship Diagram



Customer - Booking: A customer can have many reservations and can be saved in the database even if he has not yet made a booking. Each reservation must be made by a single customer.

Booking - Vehicle: A booking can have only one vehicle. If a customer needs many vehicles he will have to make many different bookings. A vehicle can be used many times and it is possible to save it in the database even if it has never been booked.

Vehicle - Model: A vehicle must have a model. Many vehicles with the same model can exist. A model can be stored even if the company does not yet own vehicles.

Booking - Booking_Employee - Employee: I used Booking_Employee to solve the many to many relationship between Booking and Employee. Every booking must have at least one employee in Booking_Employee (it is possible that there are more drivers for a booking). Each employee can be associated with many bookings in Booking_Employee. There may be employees with no bookings.

Employee - Licence: An employee can have many licences (the administrative staff must not enter their licences). Each licence must belong to an employee. When a licence expires, you can either create a new record for the new licence or update the old one based on the company's needs.

Employee - Result: An employee can have many test results because he can undergo a test several times. A result must belong to a single employee.

Result - Test: A result must have an associated test. Each type of test can have many results. It is possible that no employee has ever been subjected to a type of test.

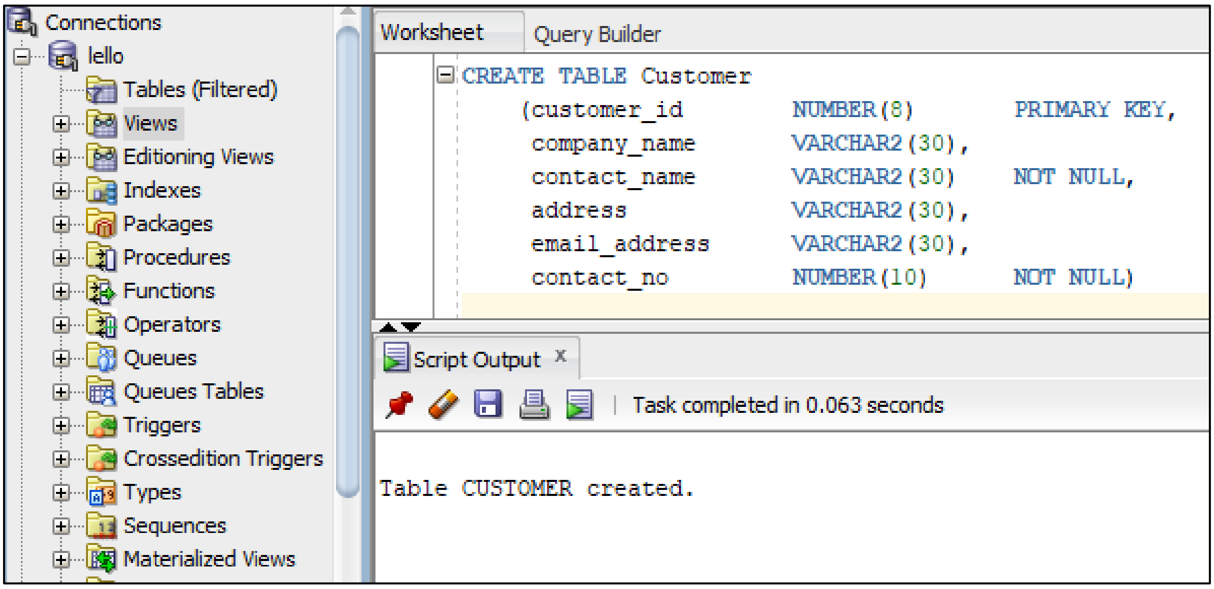
Task 2: Data Dictionary

Table	Attribute	Data Type	Length	Constraints
Customer	customer_id	number	8	PK
	company_name	varchar2	30	
	contact_name	varchar2	30	not null
	address	varchar2	30	
	email_address	varchar2	30	
	contact_no	number	10	not null
Booking	booking_id	number	8	PK
	trip_details	varchar2	50	
	booking_date	date		not null
	destination	varchar2	30	not null
	passengers_no	number	3	not null, check (passengers_no > 0)
	customer_id	number	8	FK
	registration_no	number	8	FK
Vehicle	registration_no	number	8	PK
	registration_date	date		not null
	colour	varchar2	15	
	model_id	number	8	FK
Model	model_id	number	8	PK
	model	varchar2	30	
	make	varchar2	30	

	vehicles_no	number	3	not null, check (vehicles_no > 0)
	hourly_rate	number	3	not null, check (hourly_rate > 0)
Booking_Employee	booking_id	number	8	PK, FK
	employee_id	number	8	PK, FK
Employee	employee_id	number	8	PK
	NIN	varchar2	9	not null, unique
	name	varchar2	30	not null
	address	varchar2	30	not null
	home_tel_no	number	10	not null
	DOB	date		
	gender	varchar2	1	check (m, f)
	job_description	varchar2	20	not null
	annual_salary	number	7	not null
Licence	employee_id	number	8	PK, FK
	licence_no	varchar2	16	PK
	expiry_date	date		PK
	licence_type	varchar2	20	not null, check (driver licence, PCV, CPC)
Result	employee_id	number	8	PK, FK
	test_id	number	8	PK, FK
	result_date	date		PK
	result	varchar2	20	not null
Test	test_id	number	8	PK
	type	varchar2	10	not null, check (alcohol, drug)

Task 3: CREATE TABLE statements

```
CREATE TABLE Customer
        (customer_id         NUMBER(8)          PRIMARY KEY,
         company_name        VARCHAR2(30),
         contact_name        VARCHAR2(30)       NOT NULL,
         address             VARCHAR2(30),
         email_address       VARCHAR2(30),
         contact_no          NUMBER(10)         NOT NULL)
```

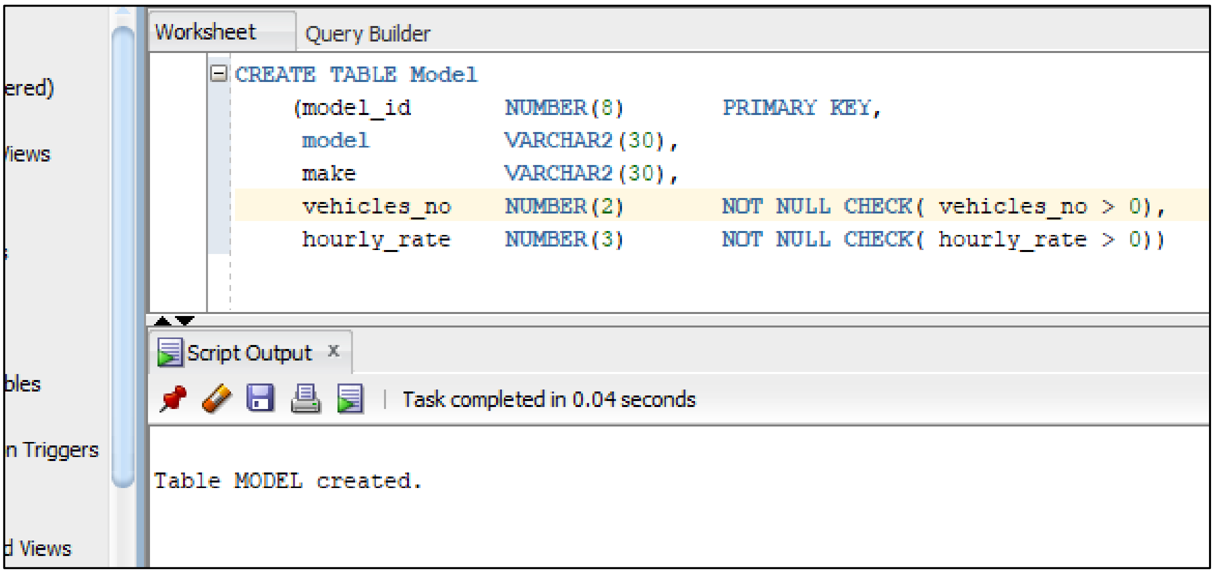


```
CREATE TABLE Customer
    (customer_id        NUMBER(8)        PRIMARY KEY,
     company_name       VARCHAR2(30),
     contact_name       VARCHAR2(30)     NOT NULL,
     address            VARCHAR2(30),
     email_address      VARCHAR2(30),
     contact_no         NUMBER(10)       NOT NULL)
```

Task completed in 0.063 seconds

```
Table CUSTOMER created.
```

```
CREATE TABLE Model
        (model_id            NUMBER(8)          PRIMARY KEY,
         model               VARCHAR2(30),
         make                VARCHAR2(30),
         vehicles_no         NUMBER(2)          NOT NULL CHECK( vehicles_no > 0),
         hourly_rate         NUMBER(3)          NOT NULL CHECK( hourly_rate > 0))
```

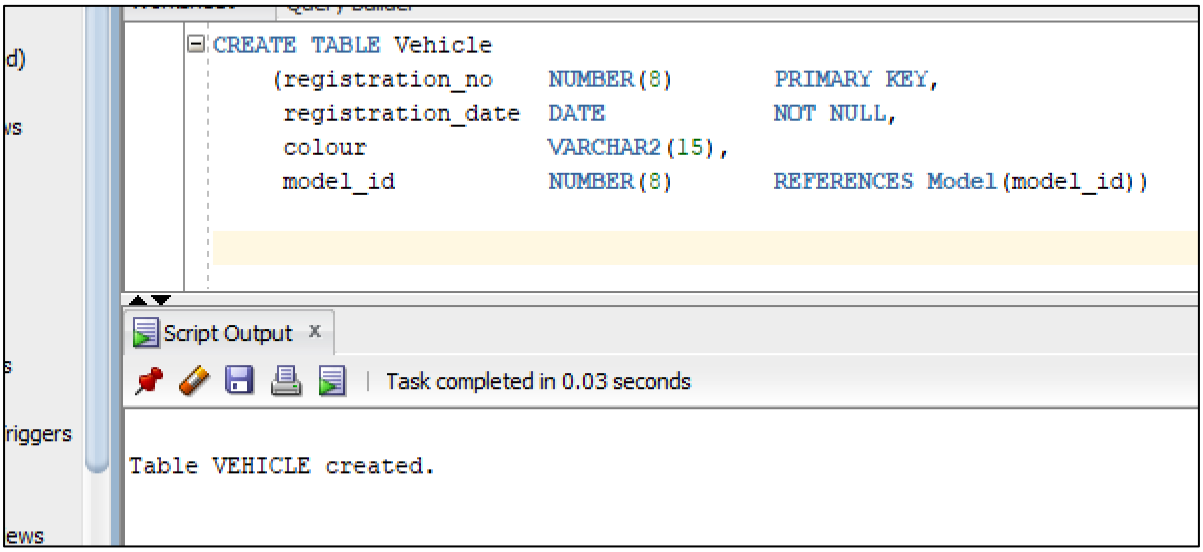


```
CREATE TABLE Model
    (model_id       NUMBER(8)       PRIMARY KEY,
     model          VARCHAR2(30),
     make           VARCHAR2(30),
     vehicles_no    NUMBER(2)       NOT NULL CHECK( vehicles_no > 0),
     hourly_rate    NUMBER(3)       NOT NULL CHECK( hourly_rate > 0))
```

Task completed in 0.04 seconds

```
Table MODEL created.
```

```
CREATE TABLE Vehicle
        (registration_no        NUMBER(8)           PRIMARY KEY,
         registration_date      DATE                NOT NULL,
         colour                 VARCHAR2(15),
         model_id               NUMBER(8)           REFERENCES Model(model_id))
```

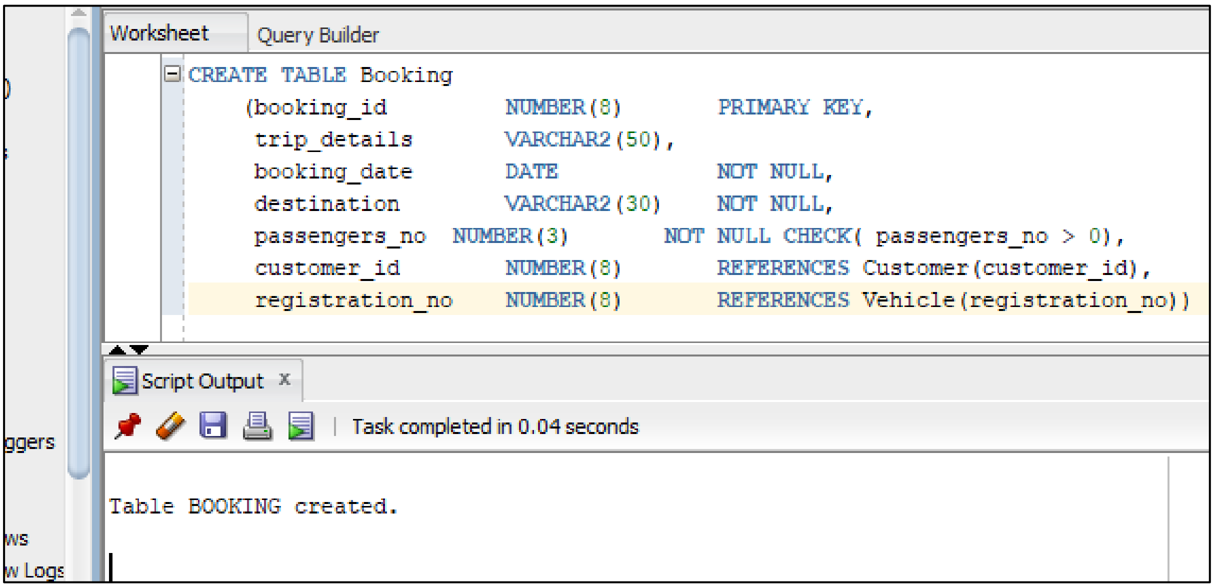


```
CREATE TABLE Vehicle
    (registration_no    NUMBER(8)       PRIMARY KEY,
     registration_date  DATE            NOT NULL,
     colour             VARCHAR2(15),
     model_id           NUMBER(8)       REFERENCES Model(model_id))
```

Task completed in 0.03 seconds

```
Table VEHICLE created.
```

```
CREATE TABLE Booking
        (booking_id             NUMBER(8)           PRIMARY KEY,
         trip_details           VARCHAR2(50),
         booking_date           DATE                NOT NULL,
         destination            VARCHAR2(30)        NOT NULL,
         passengers_no          NUMBER(3)           NOT NULL CHECK( passengers_no > 0),
         customer_id            NUMBER(8)           REFERENCES Customer(customer_id),
         registration_no        NUMBER(8)           REFERENCES Vehicle(registration_no))
```



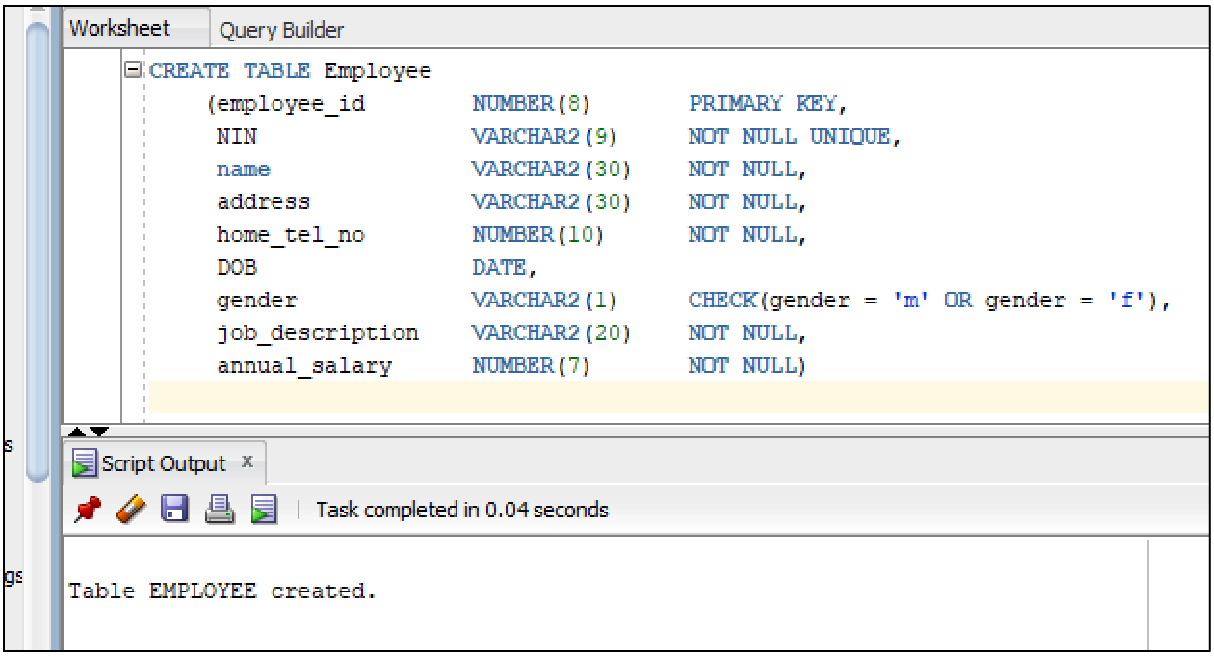
```
CREATE TABLE Booking
    (booking_id         NUMBER(8)       PRIMARY KEY,
     trip_details       VARCHAR2(50),
     booking_date       DATE            NOT NULL,
     destination        VARCHAR2(30)    NOT NULL,
     passengers_no  NUMBER(3)       NOT NULL CHECK( passengers_no > 0),
     customer_id        NUMBER(8)       REFERENCES Customer(customer_id),
     registration_no    NUMBER(8)       REFERENCES Vehicle(registration_no))
```

Task completed in 0.04 seconds

```
Table BOOKING created.
```

2018/19

```
CREATE TABLE Employee
        (employee_id        NUMBER(8)        PRIMARY KEY,
         NIN                VARCHAR2(9)      NOT NULL UNIQUE,
         name               VARCHAR2(30)     NOT NULL,
         address            VARCHAR2(30)     NOT NULL,
         home_tel_no        NUMBER(10)       NOT NULL,
         DOB                DATE,
         gender             VARCHAR2(1)      CHECK(gender = 'm' OR gender = 'f'),
         job_description    VARCHAR2(20)     NOT NULL,
         annual_salary      NUMBER(7)        NOT NULL)
```

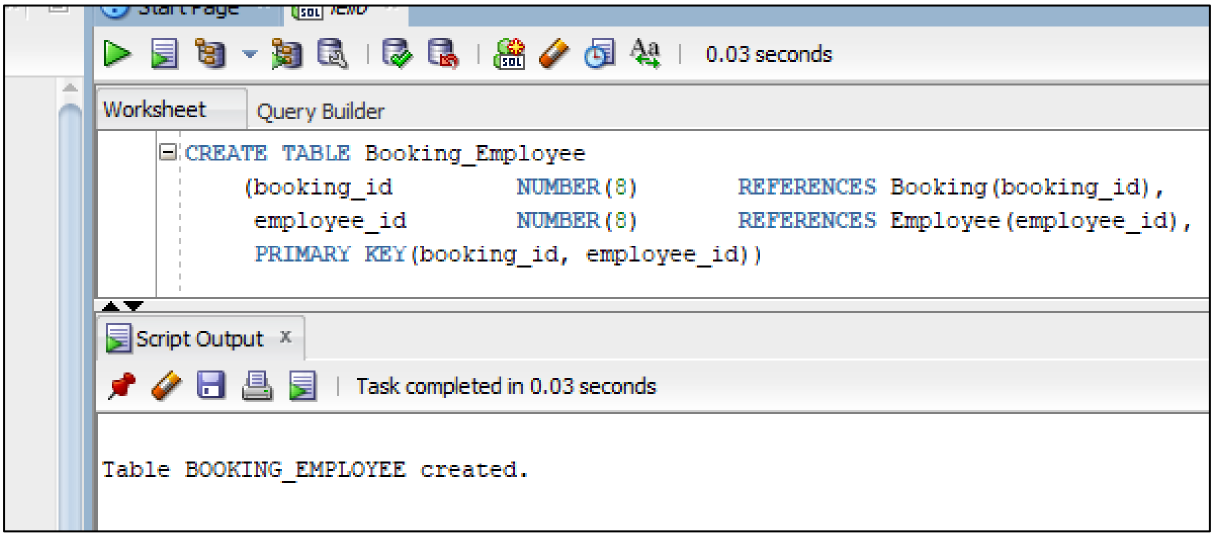


```
CREATE TABLE Employee
     (employee_id         NUMBER(8)        PRIMARY KEY,
      NIN                 VARCHAR2(9)      NOT NULL UNIQUE,
      name                VARCHAR2(30)     NOT NULL,
      address             VARCHAR2(30)     NOT NULL,
      home_tel_no         NUMBER(10)       NOT NULL,
      DOB                 DATE,
      gender              VARCHAR2(1)      CHECK(gender = 'm' OR gender = 'f'),
      job_description     VARCHAR2(20)     NOT NULL,
      annual_salary       NUMBER(7)        NOT NULL)
```

Task completed in 0.04 seconds

```
Table EMPLOYEE created.
```

```
CREATE TABLE Booking_Employee
        (booking_id         NUMBER(8)        REFERENCES Booking(booking_id),
         employee_id        NUMBER(8)        REFERENCES Employee(employee_id),
         PRIMARY KEY(booking_id, employee_id))
```

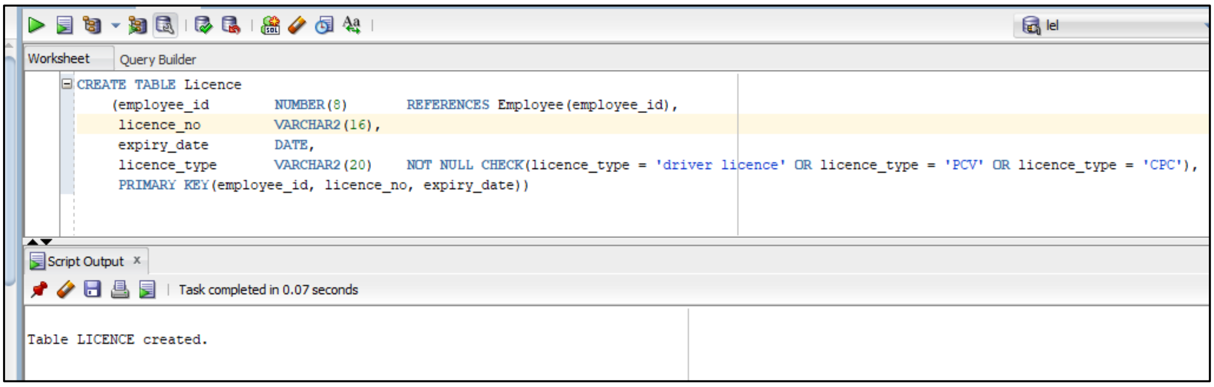


```
CREATE TABLE Booking_Employee
     (booking_id          NUMBER(8)        REFERENCES Booking(booking_id),
      employee_id         NUMBER(8)        REFERENCES Employee(employee_id),
      PRIMARY KEY(booking_id, employee_id))
```

Task completed in 0.03 seconds

```
Table BOOKING_EMPLOYEE created.
```

```
CREATE TABLE Licence
	(employee_id		NUMBER(8)		REFERENCES Employee(employee_id),
	 licence_no		VARCHAR2(16),
	 expiry_date		DATE,
	 licence_type		VARCHAR2(20)		NOT NULL CHECK(licence_type = 'driver licence' OR licence_type = 'PCV' OR licence_type = 'CPC'),
	 PRIMARY KEY(employee_id, licence_no, expiry_date))
```

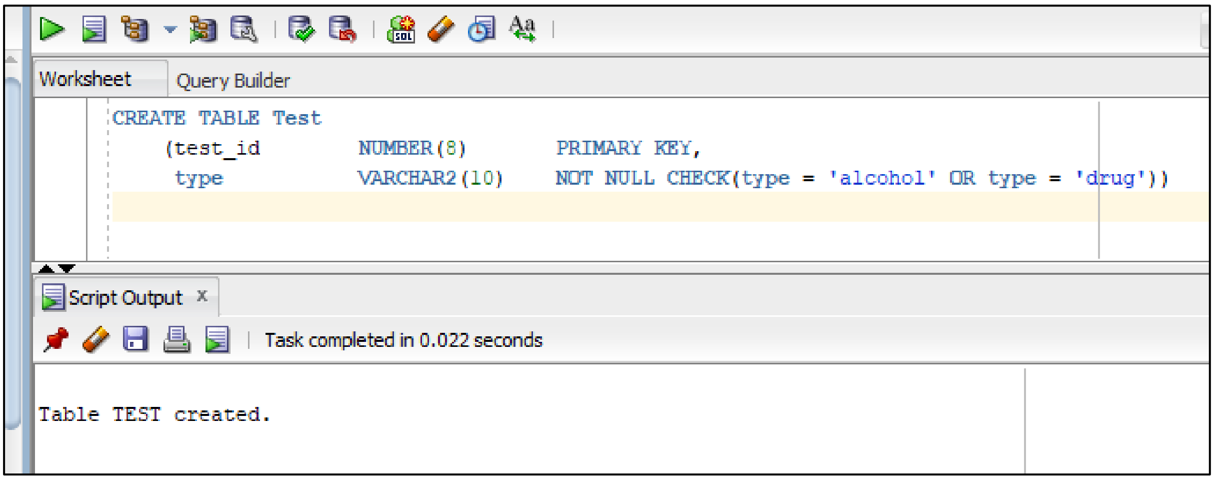


```
CREATE TABLE Licence
    (employee_id        NUMBER(8)       REFERENCES Employee(employee_id),
     licence_no         VARCHAR2(16),
     expiry_date        DATE,
     licence_type       VARCHAR2(20)    NOT NULL CHECK(licence_type = 'driver licence' OR licence_type = 'PCV' OR licence_type = 'CPC'),
     PRIMARY KEY(employee_id, licence_no, expiry_date))
```

Task completed in 0.07 seconds

```
Table LICENCE created.
```

```
CREATE TABLE Test
	(test_id		NUMBER(8)		PRIMARY KEY,
	 type			VARCHAR2(10)		NOT NULL CHECK(type = 'alcohol' OR type = 'drug'))
```

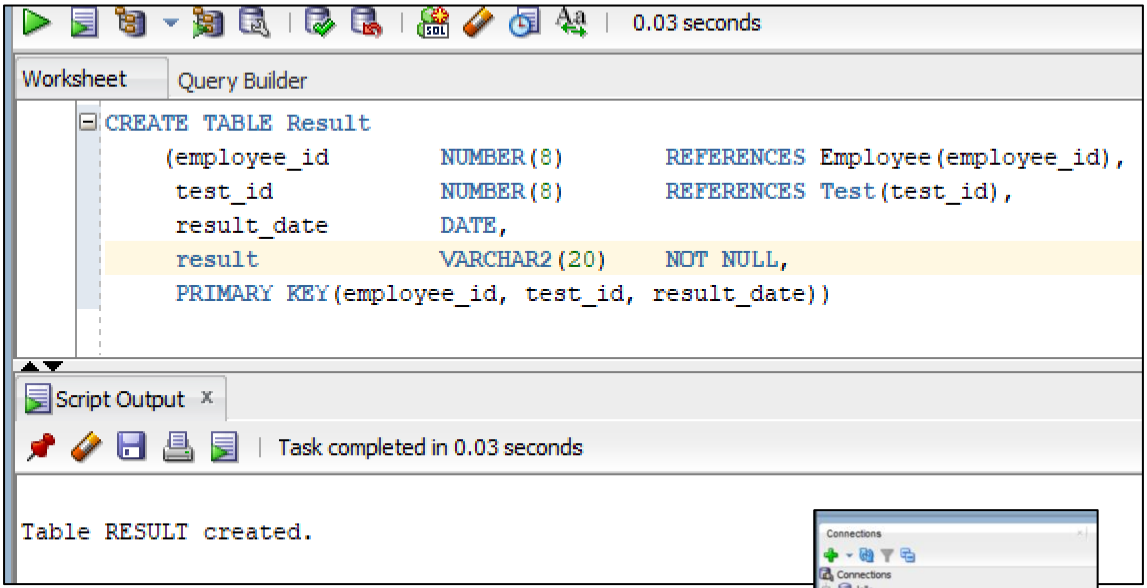


```
CREATE TABLE Test
    (test_id        NUMBER(8)       PRIMARY KEY,
     type           VARCHAR2(10)    NOT NULL CHECK(type = 'alcohol' OR type = 'drug'))
```

Task completed in 0.022 seconds

```
Table TEST created.
```

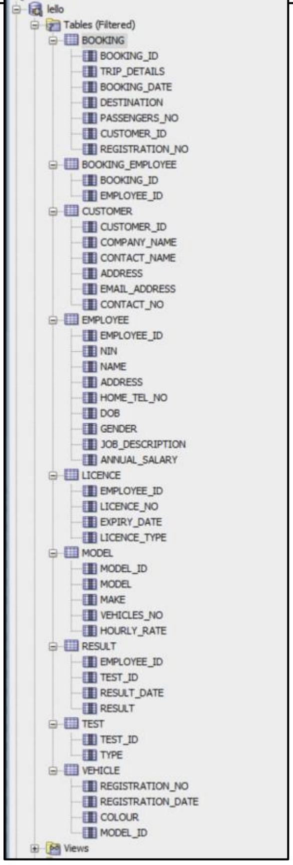
```
CREATE TABLE Result
        (employee_id         NUMBER(8)           REFERENCES Employee(employee_id),
         test_id             NUMBER(8)           REFERENCES Test(test_id),
         result_date         DATE,
         result              VARCHAR2(20)        NOT NULL,
         PRIMARY KEY(employee_id, test_id, result_date))
```



```
CREATE TABLE Result
    (employee_id        NUMBER(8)       REFERENCES Employee(employee_id),
     test_id            NUMBER(8)       REFERENCES Test(test_id),
     result_date        DATE,
     result             VARCHAR2(20)    NOT NULL,
     PRIMARY KEY(employee_id, test_id, result_date))
```

```
Table RESULT created.
```

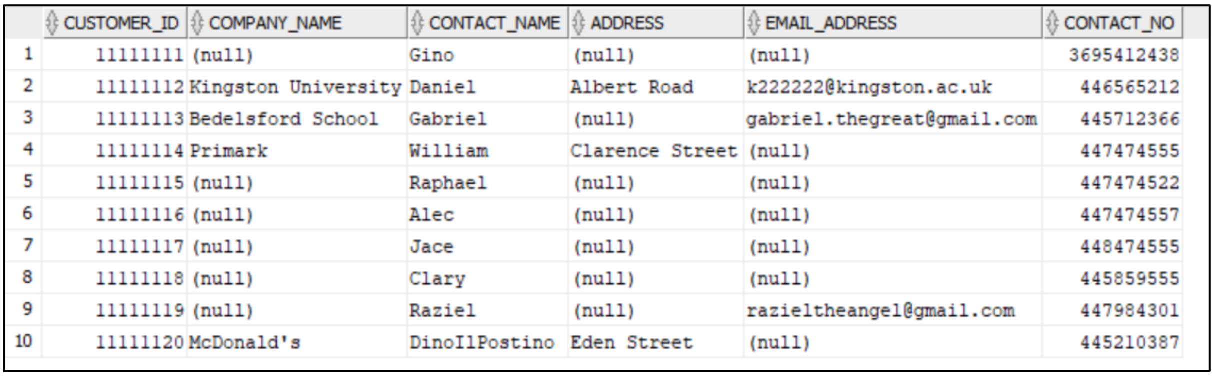
Tables:



```
Connections
Connections
  lello
    Tables (Filtered)
      BOOKING
        BOOKING_ID
        TRIP_DETAILS
        BOOKING_DATE
        DESTINATION
        PASSENGERS_NO
        CUSTOMER_ID
        REGISTRATION_NO
      BOOKING_EMPLOYEE
        BOOKING_ID
        EMPLOYEE_ID
      CUSTOMER
        CUSTOMER_ID
        COMPANY_NAME
        CONTACT_NAME
        ADDRESS
        EMAIL_ADDRESS
        CONTACT_NO
      EMPLOYEE
        EMPLOYEE_ID
        NIN
        NAME
        ADDRESS
        HOME_TEL_NO
        DOB
        GENDER
        JOB_DESCRIPTION
        ANNUAL_SALARY
      LICENCE
        EMPLOYEE_ID
        LICENCE_NO
        EXPIRY_DATE
        LICENCE_TYPE
      MODEL
        MODEL_ID
        MODEL
        MAKE
        VEHICLES_NO
        HOURLY_RATE
      RESULT
        EMPLOYEE_ID
        TEST_ID
        RESULT_DATE
        RESULT
      TEST
        TEST_ID
        TYPE
      VEHICLE
        REGISTRATION_NO
        REGISTRATION_DATE
        COLOUR
        MODEL_ID
    Views
```

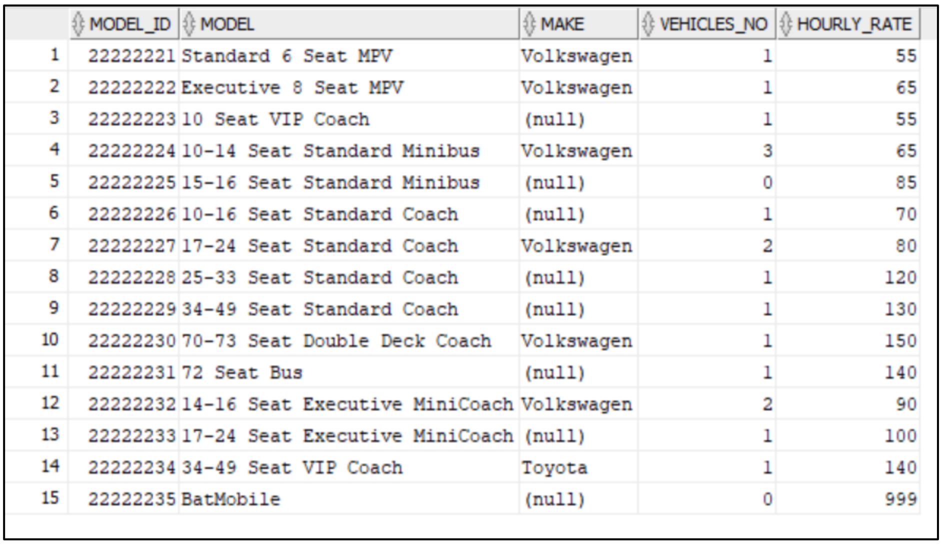
Task 4: Screenshot of populated data

Customer:



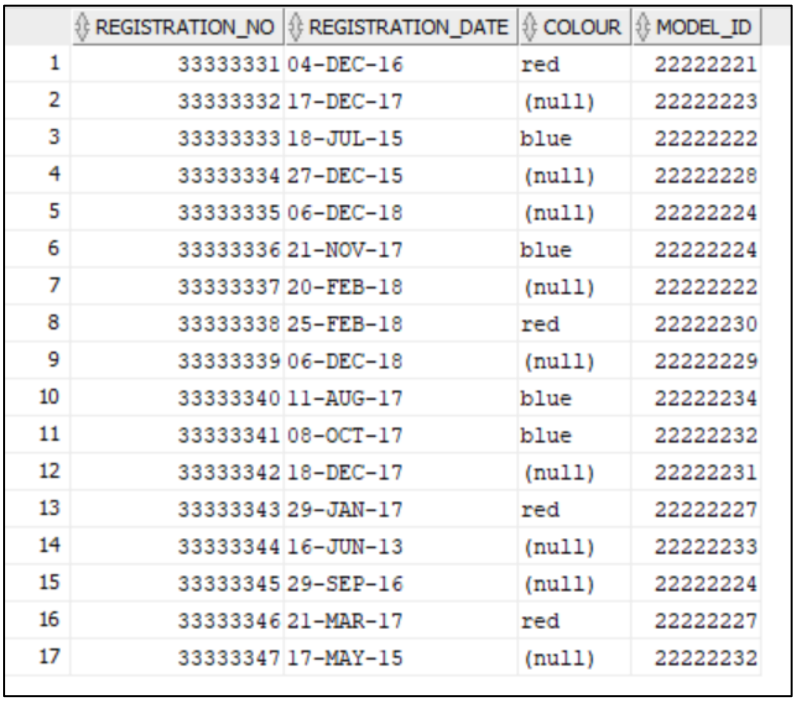
	CUSTOMER_ID	COMPANY_NAME	CONTACT_NAME	ADDRESS	EMAIL_ADDRESS	CONTACT_NO
1	11111111	(null)	Gino	(null)	(null)	3695412438
2	11111112	Kingston University	Daniel	Albert Road	k222222@kingston.ac.uk	446565212
3	11111113	Bedelsford School	Gabriel	(null)	gabriel.thegreat@gmail.com	445712366
4	11111114	Primark	William	Clarence Street	(null)	447474555
5	11111115	(null)	Raphael	(null)	(null)	447474522
6	11111116	(null)	Alec	(null)	(null)	447474557
7	11111117	(null)	Jace	(null)	(null)	448474555
8	11111118	(null)	Clary	(null)	(null)	445859555
9	11111119	(null)	Raziel	(null)	razieltheangel@gmail.com	447984301
10	11111120	McDonald's	DinoIlPostino	Eden Street	(null)	445210387

Model:



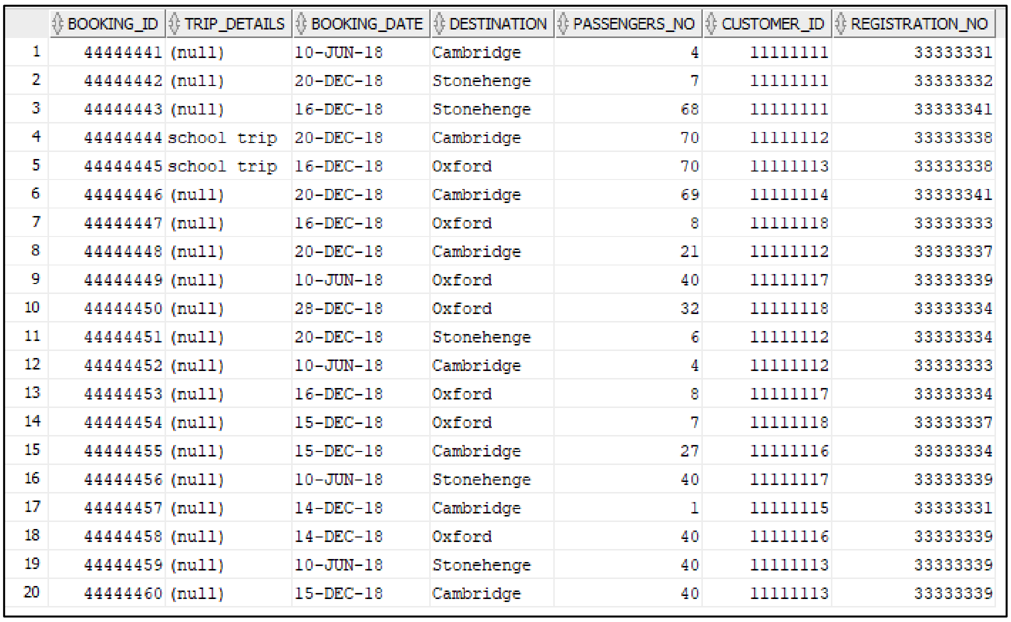
	MODEL_ID	MODEL	MAKE	VEHICLES_NO	HOURLY_RATE
1	22222221	Standard 6 Seat MPV	Volkswagen	1	55
2	22222222	Executive 8 Seat MPV	Volkswagen	1	65
3	22222223	10 Seat VIP Coach	(null)	1	55
4	22222224	10-14 Seat Standard Minibus	Volkswagen	3	65
5	22222225	15-16 Seat Standard Minibus	(null)	0	85
6	22222226	10-16 Seat Standard Coach	(null)	1	70
7	22222227	17-24 Seat Standard Coach	Volkswagen	2	80
8	22222228	25-33 Seat Standard Coach	(null)	1	120
9	22222229	34-49 Seat Standard Coach	(null)	1	130
10	22222230	70-73 Seat Double Deck Coach	Volkswagen	1	150
11	22222231	72 Seat Bus	(null)	1	140
12	22222232	14-16 Seat Executive MiniCoach	Volkswagen	2	90
13	22222233	17-24 Seat Executive MiniCoach	(null)	1	100
14	22222234	34-49 Seat VIP Coach	Toyota	1	140
15	22222235	BatMobile	(null)	0	999

Vehicle:



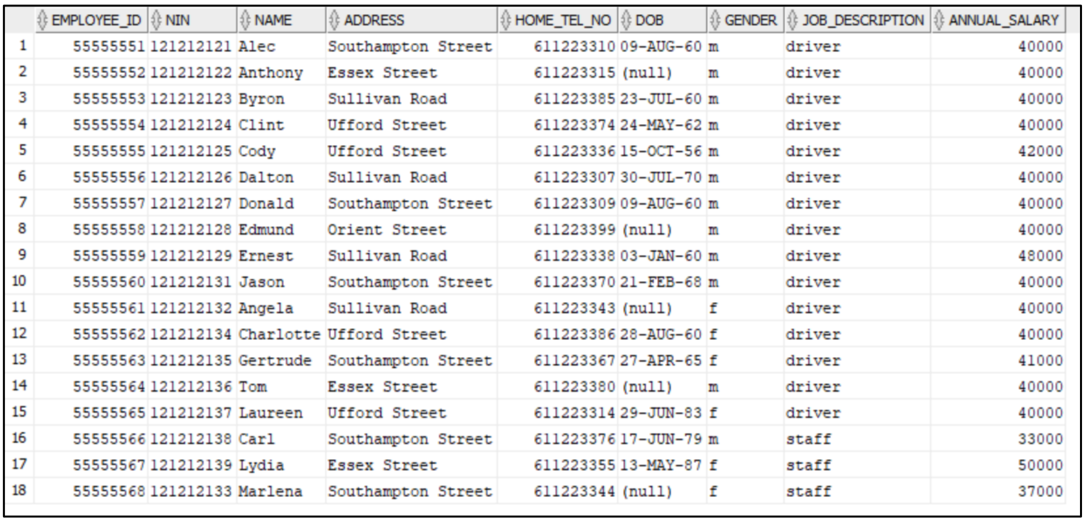
	REGISTRATION_NO	REGISTRATION_DATE	COLOUR	MODEL_ID
1	33333331	04-DEC-16	red	22222221
2	33333332	17-DEC-17	(null)	22222223
3	33333333	18-JUL-15	blue	22222222
4	33333334	27-DEC-15	(null)	22222228
5	33333335	06-DEC-18	(null)	22222224
6	33333336	21-NOV-17	blue	22222224
7	33333337	20-FEB-18	(null)	22222222
8	33333338	25-FEB-18	red	22222230
9	33333339	06-DEC-18	(null)	22222229
10	33333340	11-AUG-17	blue	22222234
11	33333341	08-OCT-17	blue	22222232
12	33333342	18-DEC-17	(null)	22222231
13	33333343	29-JAN-17	red	22222227
14	33333344	16-JUN-13	(null)	22222233
15	33333345	29-SEP-16	(null)	22222224
16	33333346	21-MAR-17	red	22222227
17	33333347	17-MAY-15	(null)	22222232

Booking:



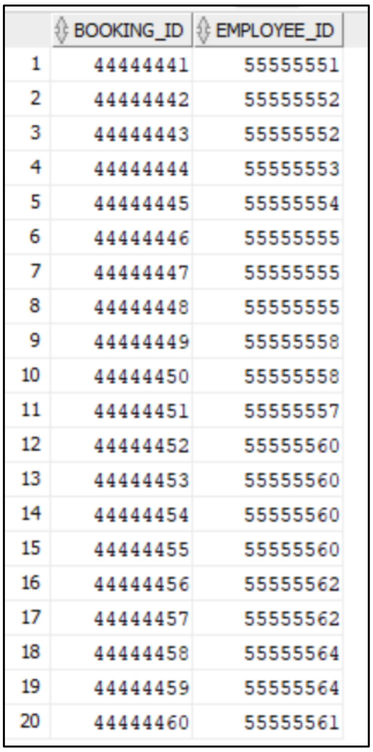
	BOOKING_ID	TRIP_DETAILS	BOOKING_DATE	DESTINATION	PASSENGERS_NO	CUSTOMER_ID	REGISTRATION_NO
1	44444441	(null)	10-JUN-18	Cambridge	4	11111111	33333331
2	44444442	(null)	20-DEC-18	Stonehenge	7	11111111	33333332
3	44444443	(null)	16-DEC-18	Stonehenge	68	11111111	33333341
4	44444444	school trip	20-DEC-18	Cambridge	70	11111112	33333338
5	44444445	school trip	16-DEC-18	Oxford	70	11111113	33333338
6	44444446	(null)	20-DEC-18	Cambridge	69	11111114	33333341
7	44444447	(null)	16-DEC-18	Oxford	8	11111118	33333333
8	44444448	(null)	20-DEC-18	Cambridge	21	11111112	33333337
9	44444449	(null)	10-JUN-18	Oxford	40	11111117	33333339
10	44444450	(null)	28-DEC-18	Oxford	32	11111118	33333334
11	44444451	(null)	20-DEC-18	Stonehenge	6	11111112	33333334
12	44444452	(null)	10-JUN-18	Cambridge	4	11111112	33333333
13	44444453	(null)	16-DEC-18	Oxford	8	11111117	33333334
14	44444454	(null)	15-DEC-18	Oxford	7	11111118	33333337
15	44444455	(null)	15-DEC-18	Cambridge	27	11111116	33333334
16	44444456	(null)	10-JUN-18	Stonehenge	40	11111117	33333339
17	44444457	(null)	14-DEC-18	Cambridge	1	11111115	33333331
18	44444458	(null)	14-DEC-18	Oxford	40	11111116	33333339
19	44444459	(null)	10-JUN-18	Stonehenge	40	11111113	33333339
20	44444460	(null)	15-DEC-18	Cambridge	40	11111113	33333339

Employee:



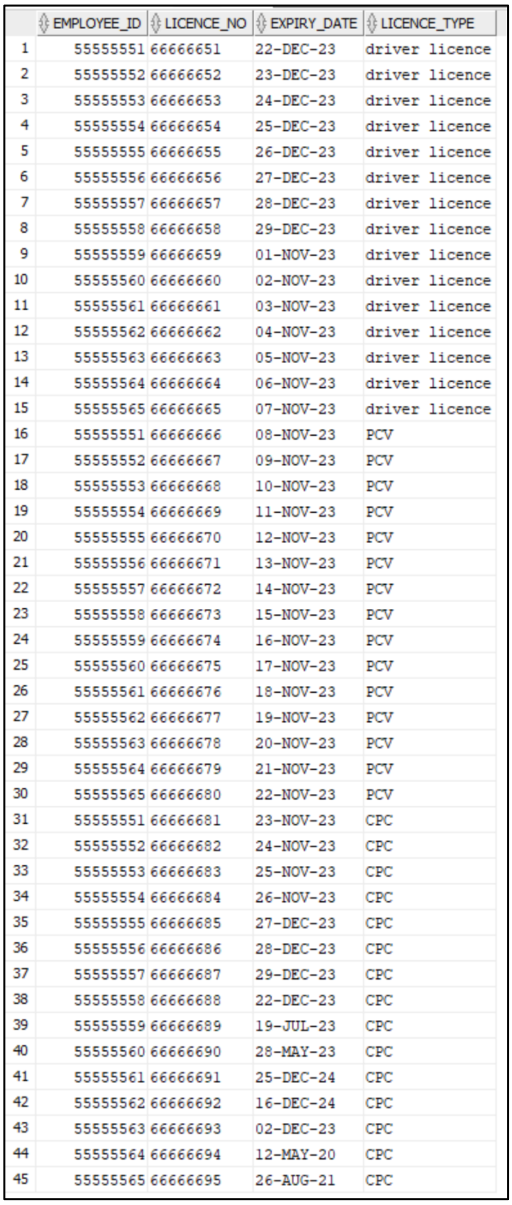
	EMPLOYEE_ID	NIN	NAME	ADDRESS	HOME_TEL_NO	DOB	GENDER	JOB_DESCRIPTION	ANNUAL_SALARY
1	55555551	121212121	Alec	Southampton Street	611223310	09-AUG-60	m	driver	40000
2	55555552	121212122	Anthony	Essex Street	611223315	(null)	m	driver	40000
3	55555553	121212123	Byron	Sullivan Road	611223385	23-JUL-60	m	driver	40000
4	55555554	121212124	Clint	Ufford Street	611223374	24-MAY-62	m	driver	40000
5	55555555	121212125	Cody	Ufford Street	611223336	15-OCT-56	m	driver	42000
6	55555556	121212126	Dalton	Sullivan Road	611223307	30-JUL-70	m	driver	40000
7	55555557	121212127	Donald	Southampton Street	611223309	09-AUG-60	m	driver	40000
8	55555558	121212128	Edmund	Orient Street	611223399	(null)	m	driver	40000
9	55555559	121212129	Ernest	Sullivan Road	611223338	03-JAN-60	m	driver	48000
10	55555560	121212131	Jason	Southampton Street	611223370	21-FEB-68	m	driver	40000
11	55555561	121212132	Angela	Sullivan Road	611223343	(null)	f	driver	40000
12	55555562	121212134	Charlotte	Ufford Street	611223386	28-AUG-60	f	driver	40000
13	55555563	121212135	Gertrude	Southampton Street	611223367	27-APR-65	f	driver	41000
14	55555564	121212136	Tom	Essex Street	611223380	(null)	m	driver	40000
15	55555565	121212137	Laureen	Ufford Street	611223314	29-JUN-83	f	driver	40000
16	55555566	121212138	Carl	Southampton Street	611223376	17-JUN-79	m	staff	33000
17	55555567	121212139	Lydia	Essex Street	611223355	13-MAY-87	f	staff	50000
18	55555568	121212133	Marlena	Southampton Street	611223344	(null)	f	staff	37000

Booking_Employee:



	BOOKING_ID	EMPLOYEE_ID
1	44444441	55555551
2	44444442	55555552
3	44444443	55555552
4	44444444	55555553
5	44444445	55555554
6	44444446	55555555
7	44444447	55555555
8	44444448	55555555
9	44444449	55555558
10	44444450	55555558
11	44444451	55555557
12	44444452	55555560
13	44444453	55555560
14	44444454	55555560
15	44444455	55555560
16	44444456	55555562
17	44444457	55555562
18	44444458	55555564
19	44444459	55555564
20	44444460	55555561

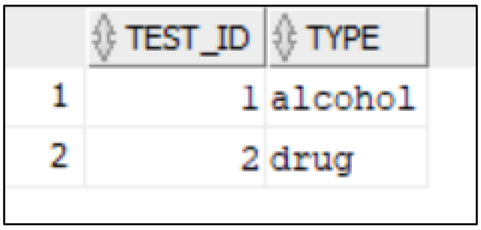
Licence:



	EMPLOYEE_ID	LICENCE_NO	EXPIRY_DATE	LICENCE_TYPE
1	55555551	66666651	22-DEC-23	driver licence
2	55555552	66666652	23-DEC-23	driver licence
3	55555553	66666653	24-DEC-23	driver licence
4	55555554	66666654	25-DEC-23	driver licence
5	55555555	66666655	26-DEC-23	driver licence
6	55555556	66666656	27-DEC-23	driver licence
7	55555557	66666657	28-DEC-23	driver licence
8	55555558	66666658	29-DEC-23	driver licence
9	55555559	66666659	01-NOV-23	driver licence
10	55555560	66666660	02-NOV-23	driver licence
11	55555561	66666661	03-NOV-23	driver licence
12	55555562	66666662	04-NOV-23	driver licence
13	55555563	66666663	05-NOV-23	driver licence
14	55555564	66666664	06-NOV-23	driver licence
15	55555565	66666665	07-NOV-23	driver licence
16	55555551	66666666	08-NOV-23	PCV
17	55555552	66666667	09-NOV-23	PCV
18	55555553	66666668	10-NOV-23	PCV
19	55555554	66666669	11-NOV-23	PCV
20	55555555	66666670	12-NOV-23	PCV
21	55555556	66666671	13-NOV-23	PCV
22	55555557	66666672	14-NOV-23	PCV
23	55555558	66666673	15-NOV-23	PCV
24	55555559	66666674	16-NOV-23	PCV
25	55555560	66666675	17-NOV-23	PCV
26	55555561	66666676	18-NOV-23	PCV
27	55555562	66666677	19-NOV-23	PCV
28	55555563	66666678	20-NOV-23	PCV
29	55555564	66666679	21-NOV-23	PCV
30	55555565	66666680	22-NOV-23	PCV
31	55555551	66666681	23-NOV-23	CPC
32	55555552	66666682	24-NOV-23	CPC
33	55555553	66666683	25-NOV-23	CPC
34	55555554	66666684	26-NOV-23	CPC
35	55555555	66666685	27-DEC-23	CPC
36	55555556	66666686	28-DEC-23	CPC
37	55555557	66666687	29-DEC-23	CPC
38	55555558	66666688	22-DEC-23	CPC
39	55555559	66666689	19-JUL-23	CPC
40	55555560	66666690	28-MAY-23	CPC
41	55555561	66666691	25-DEC-24	CPC
42	55555562	66666692	16-DEC-24	CPC
43	55555563	66666693	02-DEC-23	CPC
44	55555564	66666694	12-MAY-20	CPC
45	55555565	66666695	26-AUG-21	CPC

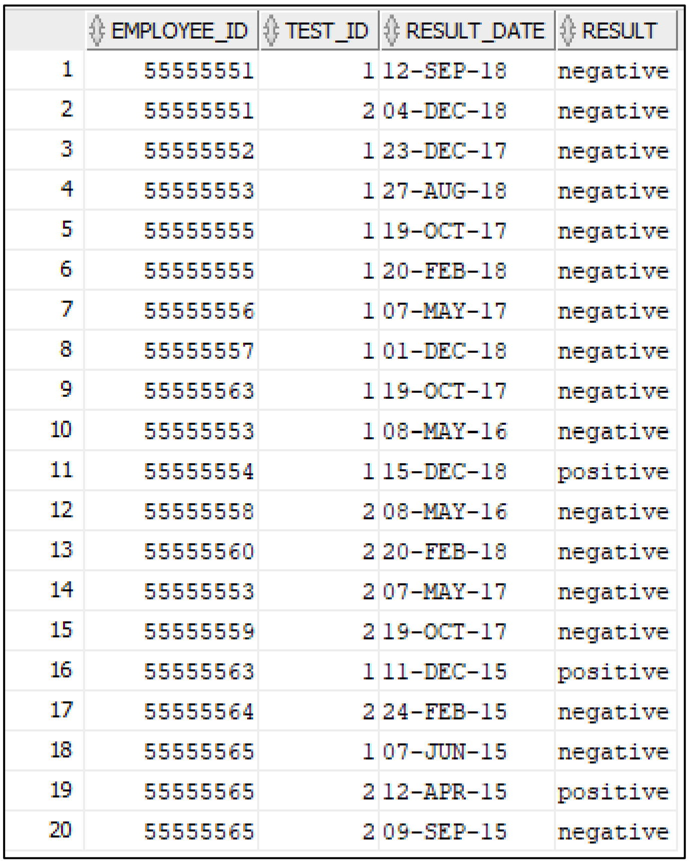
2018/19

Test:



	TEST_ID	TYPE
1	1	alcohol
2	2	drug

Result:



	EMPLOYEE_ID	TEST_ID	RESULT_DATE	RESULT
1	55555551	1	12-SEP-18	negative
2	55555551	2	04-DEC-18	negative
3	55555552	1	23-DEC-17	negative
4	55555553	1	27-AUG-18	negative
5	55555555	1	19-OCT-17	negative
6	55555555	1	20-FEB-18	negative
7	55555556	1	07-MAY-17	negative
8	55555557	1	01-DEC-18	negative
9	55555563	1	19-OCT-17	negative
10	55555553	1	08-MAY-16	negative
11	55555554	1	15-DEC-18	positive
12	55555558	2	08-MAY-16	negative
13	55555560	2	20-FEB-18	negative
14	55555553	2	07-MAY-17	negative
15	55555559	2	19-OCT-17	negative
16	55555563	1	11-DEC-15	positive
17	55555564	2	24-FEB-15	negative
18	55555565	1	07-JUN-15	negative
19	55555565	2	12-APR-15	positive
20	55555565	2	09-SEP-15	negative

Task 5: Discussion of constraints used

Customer: *customer_id* is used as the primary key to identify a customer uniquely within the table. *Contact_name* and *contact_no* have NOT NULL constraints because they are indispensable contact information. *Company_name* hasn't a NOT NULL constraint because there may be customers who do not belong to a company.

Model: *model_id* is used as the primary key to identify the model of a vehicle uniquely within the table. *Vehicles_no* and *hourly_rate* have NOT NULL constraints because they are important information for Berwyn Buses Hire Ltd to establish the availability and price of each vehicle. They also have CHECK constraints to ensure that the number entered is not negative.

Vehicle: *registration_no* is used as the primary key to identify each vehicle uniquely within the table. *Registration_date* has NOT NULL constraints because in case of breakdowns it is a useful information to know how long a vehicle has been used. *Model_id* is a foreign key referring to *model_id* in Model table, this ensures that a model entered for a vehicle exists in the model table.

Booking: *booking_id* is used as the primary key to identify each booking uniquely within the table. *Booking_date, destination* and *passengers_no* have NOT NULL constraints because they are important information to organize the trip. *Passengers_*no also has a CHECK constraint to ensure that the number entered is not negative. *Customer_id* is a foreign key referring to *customer_id* in Customer table, *registration_no* is a foreign key referring to *registration_no* in Vehicle table. This ensures that a customer and a vehicle entered for a booking exist in the customer and vehicle table.

Employee: *employee_id* is used as the primary key to identify an employee uniquely within the table. *NIN, name, address* and *home_tel_no* have NOT NULL constraints because they are indispensable contact information. Also *job_description* and *annual_salary* have NOT NULL constraints because they are useful economic information, moreover, the licenses and tests must be registered only if only if the job_description says that the employee is a driver. *NIN* has a UNIQUE constraint because no equal National Insurance Numbers can exist. Finally *gender* has a CHECK constraint to make sure that only "m" or "f" can be entered.

Booking_Employee: *booking_id* and *employee_id* form a composite primary key. *Booking_id* is a foreign key referring to *booking_id* in Booking table, *employee_id* is a foreign key referring to *employee_id* in Employee table. This ensures that a booking and an employee entered for booking_employee exist in the booking and employee table.

Licence: *employee_id, licence_no* and *expiry_date* form a composite primary key to identify a licence uniquely within the table. *Licence_type* has a NOT NULL constraint and it has a CHECK constraint to make sure that only "driver licence", “PCV” or "CPC" can be entered.

Employee_id is a foreign key referring to *employee_id* in Employee table, this ensures that a employee entered for licence exists in the employee table.

Test: *test_id* is used as the primary key to identify a test uniquely within the table. *Type* has a NOT NULL constraint and it has a CHECK constraint to make sure that only "alcohol” or "drug" can be entered.

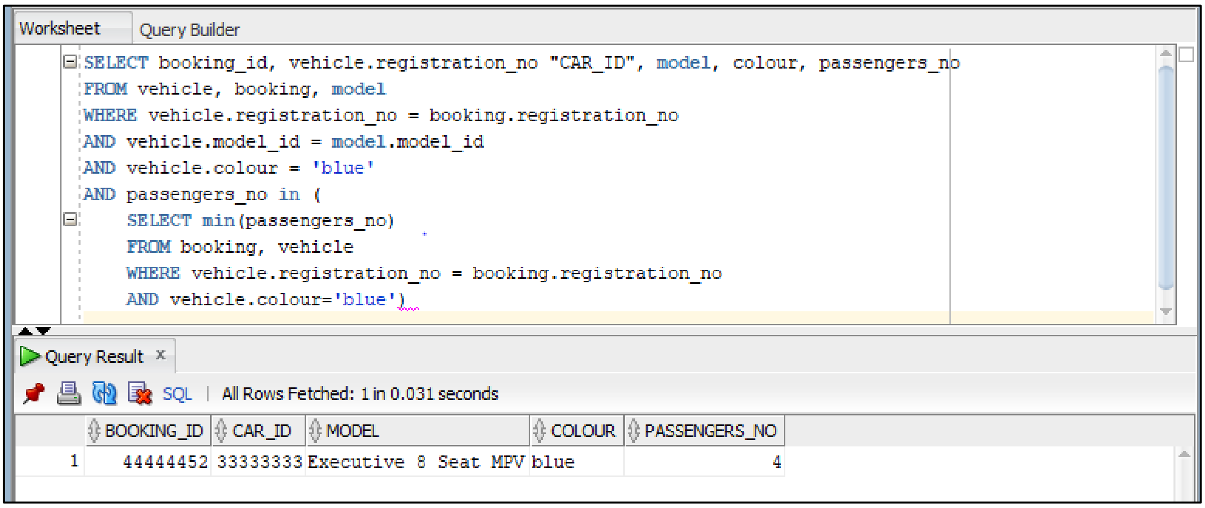
Result: *employee_id, test_id* and *result_date* form a composite primary key to identify a result uniquely within the table. *Result* has a NOT NULL constraint because it is an important information for customer safety. *Employee_id* is a foreign key referring to *employee_id* in Employee table, *test_id* is a foreign key referring to *test_id* in Test table. This ensures that a test and an employee entered for result exist in the test and employee table.

In general, every constraint is used to ensure data integrity and every foreign key is used to ensure data consistency.

Task 6 (i): Query 1

This query lists booking_id, registration_no (renamed "CAR_ID"), model, colour and passengers number of the blue vehicle/s that transported the smallest number of passengers (using a subquery):

```
SELECT booking_id, vehicle.registration_no "CAR_ID", model, colour, passengers_no
FROM vehicle, booking, model
WHERE vehicle.registration_no = booking.registration_no
AND vehicle.model_id = model.model_id
AND vehicle.colour = 'blue'
AND passengers_no in (
        SELECT min(passengers_no)
         FROM booking, vehicle
        WHERE vehicle.registration_no = booking.registration_no
        AND vehicle.colour='blue')
```



```
SELECT booking_id, vehicle.registration_no "CAR_ID", model, colour, passengers_no
FROM vehicle, booking, model
WHERE vehicle.registration_no = booking.registration_no
AND vehicle.model_id = model.model_id
AND vehicle.colour = 'blue'
AND passengers_no in (
    SELECT min(passengers_no)
    FROM booking, vehicle
    WHERE vehicle.registration_no = booking.registration_no
    AND vehicle.colour='blue')
```

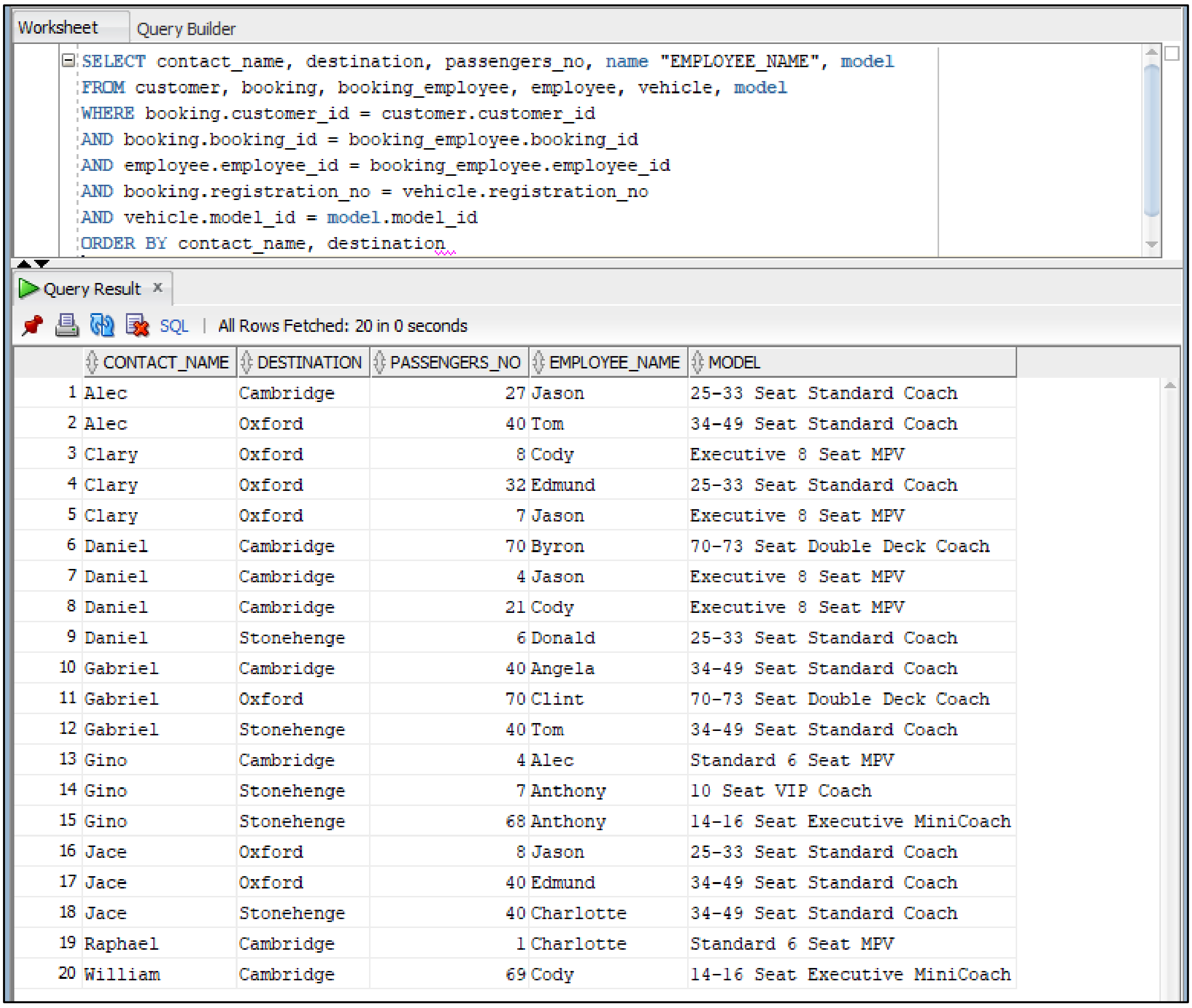
All Rows Fetched: 1 in 0.031 seconds

```
     BOOKING_ID  CAR_ID   MODEL                COLOUR  PASSENGERS_NO
  1    44444452 33333333 Executive 8 Seat MPV blue                 4
```

Task 6 (ii): Query 2

This query lists customer name, destination, passengers' number, employee name and model of each booking. I made this query to show the correct connection between multiple tables:

SELECT contact_name, destination, passengers_no, name "EMPLOYEE_NAME", model
FROM customer, booking, booking_employee, employee, vehicle, model
WHERE booking.customer_id = customer.customer_id
AND booking.booking_id = booking_employee.booking_id
AND employee.employee_id = booking_employee.employee_id
AND booking.registration_no = vehicle.registration_no
AND vehicle.model_id = model.model_id
ORDER BY contact_name, destination



```
SELECT contact_name, destination, passengers_no, name "EMPLOYEE_NAME", model
FROM customer, booking, booking_employee, employee, vehicle, model
WHERE booking.customer_id = customer.customer_id
AND booking.booking_id = booking_employee.booking_id
AND employee.employee_id = booking_employee.employee_id
AND booking.registration_no = vehicle.registration_no
AND vehicle.model_id = model.model_id
ORDER BY contact_name, destination
```

All Rows Fetched: 20 in 0 seconds

	CONTACT_NAME	DESTINATION	PASSENGERS_NO	EMPLOYEE_NAME	MODEL
1	Alec	Cambridge	27	Jason	25-33 Seat Standard Coach
2	Alec	Oxford	40	Tom	34-49 Seat Standard Coach
3	Clary	Oxford	8	Cody	Executive 8 Seat MPV
4	Clary	Oxford	32	Edmund	25-33 Seat Standard Coach
5	Clary	Oxford	7	Jason	Executive 8 Seat MPV
6	Daniel	Cambridge	70	Byron	70-73 Seat Double Deck Coach
7	Daniel	Cambridge	4	Jason	Executive 8 Seat MPV
8	Daniel	Cambridge	21	Cody	Executive 8 Seat MPV
9	Daniel	Stonehenge	6	Donald	25-33 Seat Standard Coach
10	Gabriel	Cambridge	40	Angela	34-49 Seat Standard Coach
11	Gabriel	Oxford	70	Clint	70-73 Seat Double Deck Coach
12	Gabriel	Stonehenge	40	Tom	34-49 Seat Standard Coach
13	Gino	Cambridge	4	Alec	Standard 6 Seat MPV
14	Gino	Stonehenge	7	Anthony	10 Seat VIP Coach
15	Gino	Stonehenge	68	Anthony	14-16 Seat Executive MiniCoach
16	Jace	Oxford	8	Jason	25-33 Seat Standard Coach
17	Jace	Oxford	40	Edmund	34-49 Seat Standard Coach
18	Jace	Stonehenge	40	Charlotte	34-49 Seat Standard Coach
19	Raphael	Cambridge	1	Charlotte	Standard 6 Seat MPV
20	William	Cambridge	69	Cody	14-16 Seat Executive MiniCoach

Task 6 (iii): Query 3

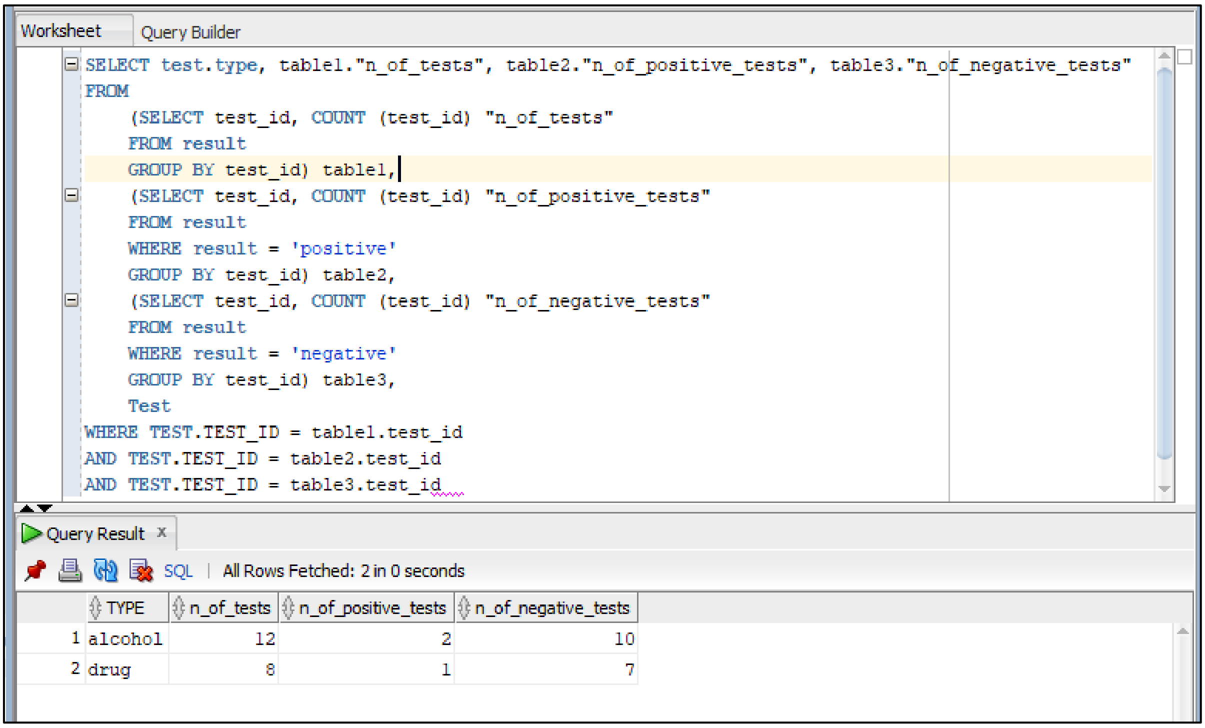
Finally, I made this query. It lists the type of a test, how many tests of that type were done to the drivers, how many were positive and how many were negative. I made this query to demonstrate the use of the group function COUNT:

```
SELECT test.type, table1."n_of_tests", table2."n_of_positive_tests",
table3."n_of_negative_tests"
FROM
        (SELECT test_id, COUNT (test_id) "n_of_tests"
        FROM result
        GROUP BY test_id) table1,

        (SELECT test_id, COUNT (test_id) "n_of_positive_tests"
        FROM result
        WHERE result = 'positive'
        GROUP BY test_id) table2,

        (SELECT test_id, COUNT (test_id) "n_of_negative_tests"
        FROM result
        WHERE result = 'negative'
        GROUP BY test_id) table3,

        Test
WHERE TEST.TEST_ID = table1.test_id
AND TEST.TEST_ID = table2.test_id
AND TEST.TEST_ID = table3.test_id
```



```
SELECT test.type, table1."n_of_tests", table2."n_of_positive_tests", table3."n_of_negative_tests"
FROM
    (SELECT test_id, COUNT (test_id) "n_of_tests"
    FROM result
    GROUP BY test_id) table1,
    (SELECT test_id, COUNT (test_id) "n_of_positive_tests"
    FROM result
    WHERE result = 'positive'
    GROUP BY test_id) table2,
    (SELECT test_id, COUNT (test_id) "n_of_negative_tests"
    FROM result
    WHERE result = 'negative'
    GROUP BY test_id) table3,
    Test
WHERE TEST.TEST_ID = table1.test_id
AND TEST.TEST_ID = table2.test_id
AND TEST.TEST_ID = table3.test_id
```

All Rows Fetched: 2 in 0 seconds

	TYPE	n_of_tests	n_of_positive_tests	n_of_negative_tests
1	alcohol	12	2	10
2	drug	8	1	7