

CI5320 Database Application Development

Coursework 1 -

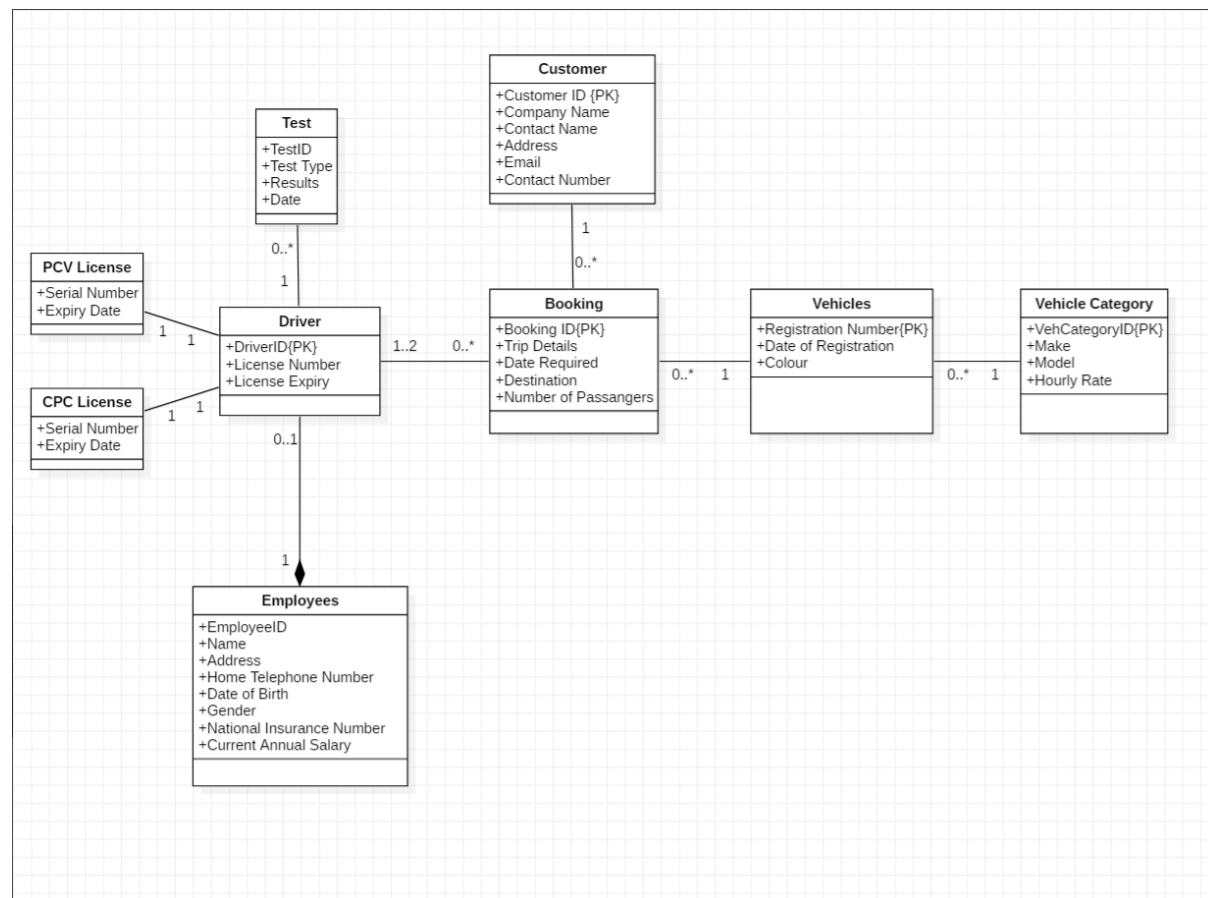
Name: Tomasz Przybylski

K Number: K1602155

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

Booking HAS_A Customer, Customer MAY_HAVE_MANY Bookings – One to Many
 Booking HAS_A Vehicle, Vehicles MAY_HAVE_MANY Bookings – One To Many
 Vehicle HAS_A Vehicle Category, Vehicle Category HAS_MANY Vehicles – One To Many
 Driver MAY_HAVE_MANY Bookings, Bookings HAS_ONE-TWO Drivers - One To Many
 Driver MAY_HAVE_MANY Tests, Test HAS_A Driver – One To Many
 Driver HAS_A PCV License, PCV License HAS_A Driver – One To One
 Driver HAS_A CPC License, CPC License HAS_A Driver – One To One
 Driver IS_AN Employee, Employee MAY_BE_A Driver – One To Many



Task 2: Data Dictionary

	A	B	C	D	E	F	G
1	Relation Name	Attribute Name	Data Type	Length	PK/FK	NOT NULL?	Other Comments
2	Customer	Customer_ID	VARCHAR2	8	PK	NOT NULL	
3		Company_Name	VARCHAR2	300		NOT NULL	
4		Contact_Name	VARCHAR2	300		NOT NULL	
5		Address	VARCHAR2	300		NOT NULL	
6		Email	VARCHAR2	30			CHECK (Email = NOT NULL OR Contact_Number = NOT NULL)
7		Contact_Number	CHAR	11			CHECK (Email = NOT NULL OR Contact_Number = NOT NULL)
8							
9	Booking	Booking_ID	VARCHAR2	8	PK	NOT NULL	
10		Trip_Details	VARCHAR2	30			
11		Date_Required	DATE			NOT NULL	
12		Destination	VARCHAR2	30		NOT NULL	
13		Number_of_Passangers	NUMBER	3		NOT NULL	CHECK (Number_of_Passangers >=0)
14		Customer_ID	VARCHAR2	8	FK	NOT NULL	
15		Driver_ID	VARCHAR2	8	FK	NOT NULL	
16		Registration_Number	VARCHAR2	8	FK	NOT NULL	
17							
18	Vehicles	Registration_Number	VARCHAR2	9	PK	NOT NULL	
19		Date_of_Registration	DATE			NOT NULL	
20		Colour	VARCHAR2	30			
21		Veh_Category_ID	VARCHAR2	8	FK	NOT NULL	
22							
23	Vehicle_Category	Veh_Category_ID	VARCHAR2	8	PK	NOT NULL	
24		VMake	VARCHAR2	24		NOT NULL	
25		VModel	VARCHAR2	24		NOT NULL	
26		Hourly_Rate	NUMBER	*,2		NOT NULL	
27							
28	Driver	Driver_ID	VARCHAR2	8	PK	NOT NULL	
29		License_Number	VARCHAR2	16		NOT NULL	
30		License_Expiry	DATE			NOT NULL	
31		PCV_Serial_Number	VARCHAR2	8	FK	NOT NULL	
32		CPC_Serial_Number	VARCHAR2	8	FK	NOT NULL	
33		Employee_ID	VARCHAR2	8	FK	NOT NULL	
34							
35	TestT	Test_ID	VARCHAR2	8	PK	NOT NULL	
36		Test_Type	VARCHAR2	8		NOT NULL	
37		Results	VARCHAR2	30		NOT NULL	
38		TDate	DATE			NOT NULL	DEFAULT sysdate
39		Driver_ID	VARCHAR2	8	FK	NOT NULL	
40							
41	PCV_License	PCV_Serial_Number	VARCHAR2	8	PK	NOT NULL	
42		PCV_Expiry_Date	DATE			NOT NULL	
43							
44	CPC_License	CPC_Serial_Number	VARCHAR2	8	PK	NOT NULL	
45		CPC_Expiry_Date	DATE			NOT NULL	
46							
47	Employees	Employee_ID	VARCHAR2	8	PK	NOT NULL	
48		ENAME	VARCHAR2	30		NOT NULL	
49		Address	VARCHAR2	30		NOT NULL	
50		Home_Telephone_Number	CHAR	11		NOT NULL	
51		Job_Description	VARCHAR2	300			
52		Date_of_Birth	DATE			NOT NULL	
53		Gender	CHAR	1		NOT NULL	CHECK (Gender ='M' OR Gender = 'F')
54		National_Insurance_Number	VARCHAR2	14		NOT NULL	
55		Current_Annual_Salary	NUMBER	8		NOT NULL	CHECK (Current_Annual_Salary > 0)

Task 3: CREATE TABLE statements

```
CREATE TABLE Employee
(Employee_ID varchar2(8) PRIMARY KEY,
EName varchar2(24) NOT NULL,
Address varchar2(30) NOT NULL,
Home_Telephone_Number char(11) NOT NULL,
Date_Of_Birth date NOT NULL,
Gender char(1) CHECK (Gender = "M" OR Gender = "F") NOT NULL,
Job_Description varchar2(300),
National_Insurance_Number varchar2(14) NOT NULL,
Current_Annual_Salary number(8) CHECK(Current_Annual_Salary > 0) NOT NULL)
```

```
CREATE TABLE Driver
(Driver_ID varchar2(8) PRIMARY KEY,
License_Number varchar2(16) UNIQUE NOT NULL,
License_Expiry date NOT NULL,
PCV_Serial_Number varchar2(8) REFERENCES PCV_License(PCV_Serial_Number) NOT NULL,
CPC_Serial_Number varchar2(8) REFERENCES CPC_License(CPC_Serial_Number) NOT NULL,
Employee_ID varchar2(8) REFERENCES Employee(Employee_ID) NOT NULL)
```

```
CREATE TABLE TestT
(Test_ID varchar2(8) PRIMARY KEY,
Test_Type varchar2(8) NOT NULL,
Results varchar2(8) NOT NULL,
TDate date DEFAULT sysdate,
Driver_ID varchar2(8) REFERENCES Driver(Driver_ID) NOT NULL)
```

```
CREATE TABLE PCV_License
(PCV_Serial_Number varchar2(8) PRIMARY KEY,
PCV_Expiry_Date date NOT NULL)
```

```
CREATE TABLE CPC_License
(CPC_Serial_Number varchar2(8) PRIMARY KEY,
CPC_Expiry_Date date NOT NULL)
```

```
CREATE TABLE Vehicle
(Registration_Number varchar2(9) PRIMARY KEY,
Date_Of_Registration date NOT NULL,
Colour varchar2(30),
Veh_Category_ID varchar2(8) REFERENCES Vehicle_Category(Veh_Category_ID) NOT NULL)
```

```
CREATE TABLE Vehicle_Category
(Veh_Category_ID varchar2(8) PRIMARY KEY,
VMake varchar2(24) NOT NULL,
VModel varchar2(24) NOT NULL,
Hourly_Rate number(*,2) NOT NULL)
```

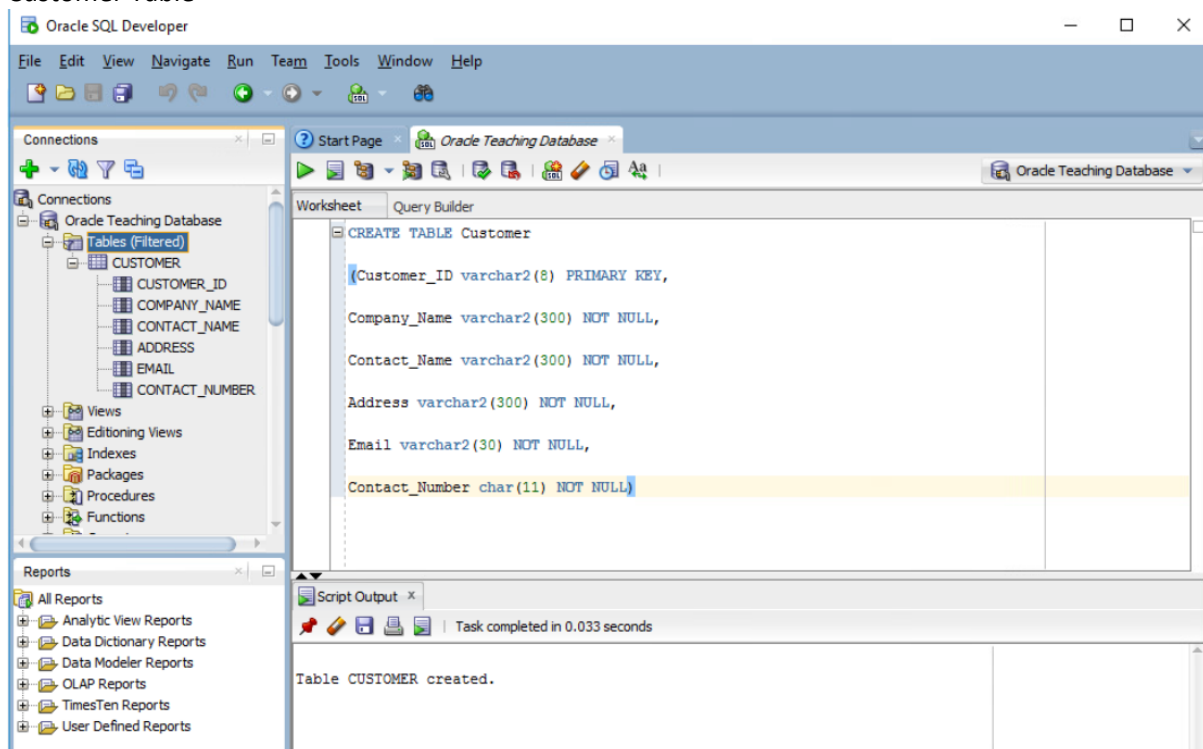
```
CREATE TABLE Booking
(Booking_ID varchar2(8) PRIMARY KEY,
Trip_Details varchar2(200),
```

2018/19

*Date_Required date NOT NULL,
Destination varchar2(300) NOT NULL,
Number_Of_Passangers number(3) CHECK(Number_Of_Passangers >=0) NOT NULL,
Customer_ID varchar2(8) REFERENCES Customer(Customer_ID) NOT NULL,
Driver_ID varchar2(8) REFERENCES Driver(Driver_ID) NOT NULL,
Registration_Number varchar2(8) REFERENCES Vehicle(Registration_Number) NOT NULL)*

*CREATE TABLE Customer
(Customer_ID varchar2(8) PRIMARY KEY,
Company_Name varchar2(300) NOT NULL,
Contact_Name varchar2(300) NOT NULL,
Address varchar2(300) NOT NULL,
Email varchar2(30) NOT NULL,
Contact_Number char(11) NOT NULL)*

Customer Table



2018/19

Vehicle_Category Table

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'Oracle Teaching Database' with a tree view of tables including 'CUSTOMER', 'VEHICLE_CATEGORY', 'VEH_CATEGORY_ID', 'VMAKE', 'VMODEL', and 'HOURLY_RATE'. The 'Worksheet' pane displays the following SQL code:

```
CREATE TABLE Vehicle_Category  
(Veh_Category_ID varchar2(8) PRIMARY KEY,  
VMake varchar2(24) NOT NULL,  
VModel varchar2(24) NOT NULL,  
Hourly_Rate number(*,2) NOT NULL)
```

The 'Script Output' pane at the bottom shows the execution results:

```
Table CUSTOMER created.  
  
Table VEHICLE_CATEGORY created.
```

Vehicle Table

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Connections' pane shows the 'Oracle Teaching Database' with a tree view of tables including 'CUSTOMER', 'VEHICLE', 'REGISTRATION_NUM', 'DATE_OF_REGISTRA', 'COLOUR', 'VEH_CATEGORY_ID', and 'VEHICLE_CATEGORY'. The 'Worksheet' pane displays the following SQL code:

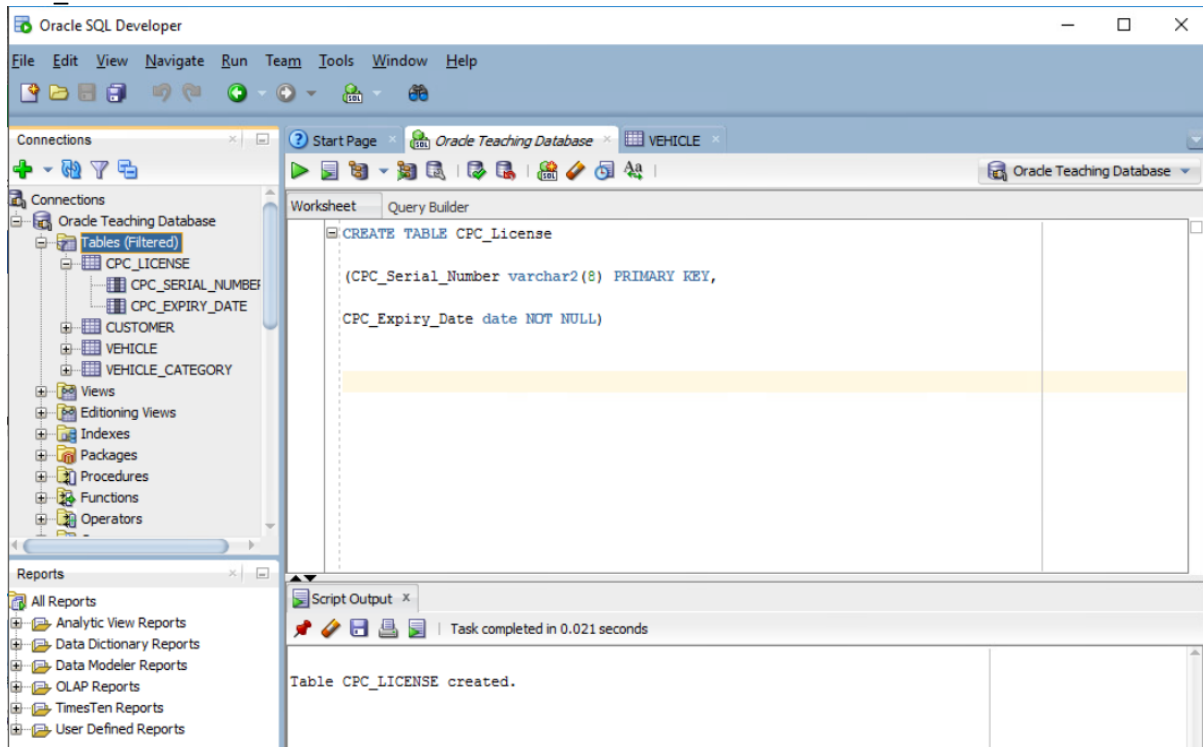
```
CREATE TABLE Vehicle  
(Registration_Number varchar2(9) PRIMARY KEY,  
Date_Of_Registration date NOT NULL,  
Colour varchar2(30),  
Veh_Category_ID varchar2(8) REFERENCES Vehicle_Category(Veh_Category_ID) NOT NULL)
```

The 'Script Output' pane at the bottom shows the execution results:

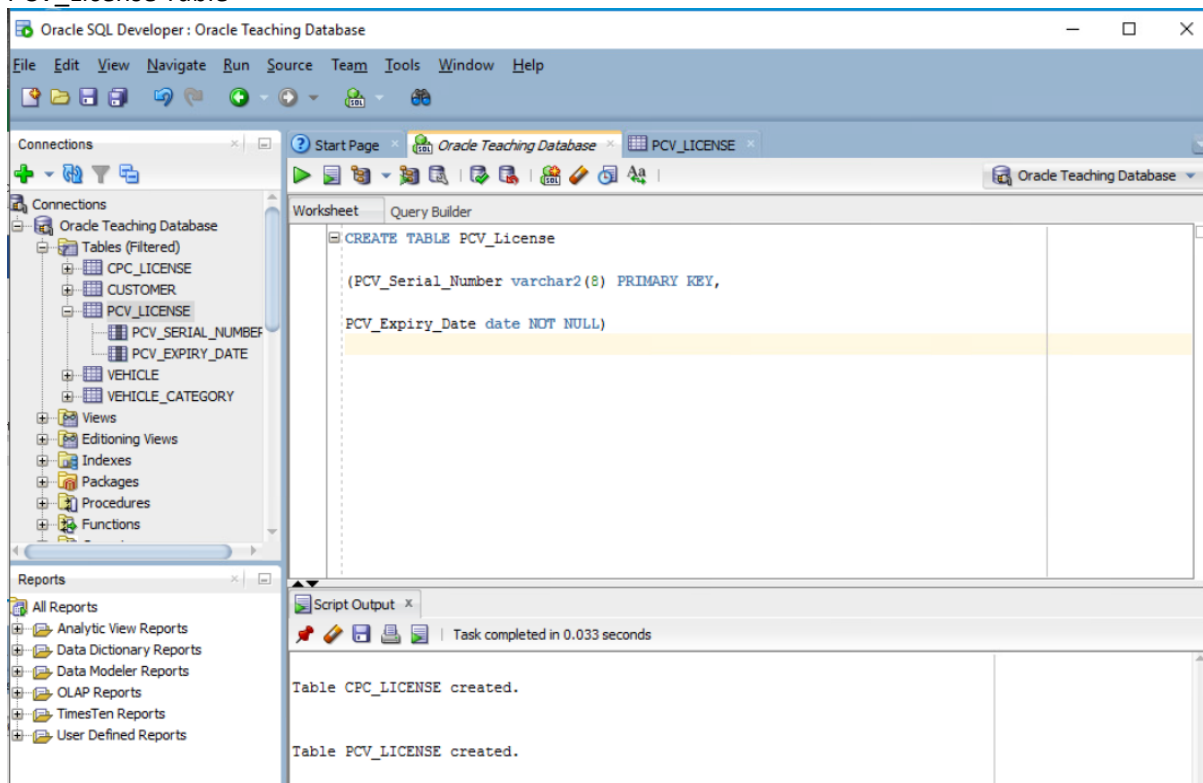
```
Table CUSTOMER created.  
  
Table VEHICLE_CATEGORY created.  
  
Table VEHICLE created.
```

2018/19

CPC_License

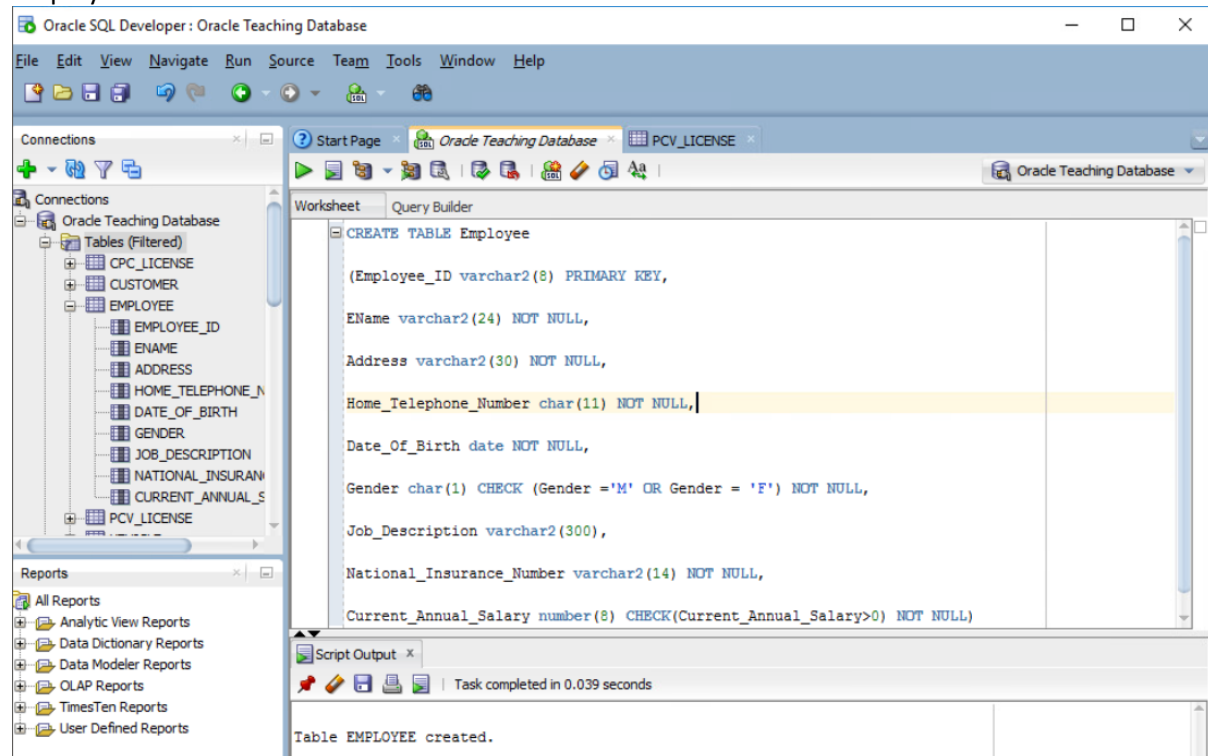


PCV_License Table

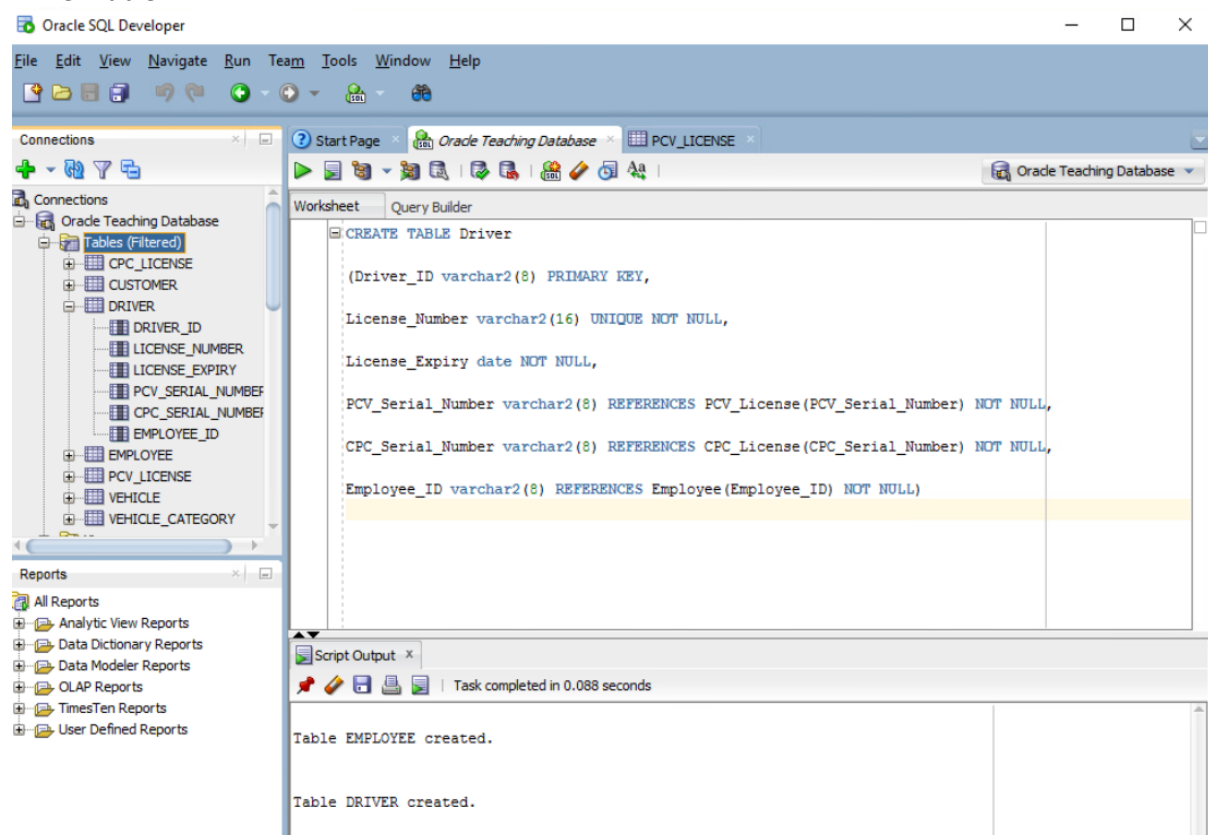


2018/19

Employee Table



Driver Table



2018/19

TestT Table

The screenshot displays the Oracle SQL Developer interface with the 'Oracle Teaching Database' connected. The left-hand 'Connections' pane shows a tree view of the database schema, including tables like CPC_LICENSE, CUSTOMER, DRIVER, EMPLOYEE, PCV_LICENSE, VEHICLE, and VEHICLE_CATEGORY. The main 'Worksheet' area contains the following SQL code to create the 'TestT' table:

```
CREATE TABLE TestT
(
  Test_ID varchar2(8) PRIMARY KEY,
  Test_Type varchar2(8) NOT NULL,
  Results varchar2(8) NOT NULL,
  TDate date DEFAULT sysdate,
  Driver_ID varchar2(8) REFERENCES Driver(Driver_ID) NOT NULL
)
```

Below the SQL editor, the 'Script Output' pane shows the execution results, indicating that the tables EMPLOYEE, DRIVER, and TESTT were successfully created.

Table EMPLOYEE created.

Table DRIVER created.

Table TESTT created.

Booking Table

The screenshot shows the Oracle SQL Developer interface with the 'Oracle Teaching Database' connected. The left pane displays the 'Connections' tree with 'Oracle Teaching Database' selected, showing a list of tables including CPC_LICENSE, CUSTOMER, DRIVER, EMPLOYEE, PCV_LICENSE, TEST, VEHICLE, and VEHICLE_CATEGORY. The main workspace is in 'Query Builder' mode, showing a SQL script to create a 'Booking' table. The script defines columns: Booking_ID (PRIMARY KEY), Trip_Details, Date_Required, Destination, Number_Of_Passangers, Customer_ID, Driver_ID, and Registration_Number, with various constraints. The 'Script Output' pane at the bottom shows the execution results, including an error report for 'ORA-00910: specified length too long for its datatype' and a confirmation that the 'Table BOOKING created'.

Oracle SQL Developer : Oracle Teaching Database

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Teaching Database

Tables (Filtered)

- CPC_LICENSE
- CUSTOMER
- DRIVER
- EMPLOYEE
- PCV_LICENSE
- TEST
- VEHICLE
- VEHICLE_CATEGORY

Views

- Editing Views
- Indexes
- Packages
- Procedures

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Worksheet

Query Builder

```
CREATE TABLE Booking
(Booking_ID varchar2(8) PRIMARY KEY,
Trip_Details varchar2(200),
Date_Required date NOT NULL,
Destination varchar2(300) NOT NULL,
Number_Of_Passangers number(3) CHECK(Number_Of_Passangers >=0) NOT NULL,
Customer_ID varchar2(8) REFERENCES Customer(Customer_ID),
Driver_ID varchar2(8) REFERENCES Driver(Driver_ID),
Registration_Number varchar2(8) REFERENCES Vehicle(Registration_Number))
```

Script Output

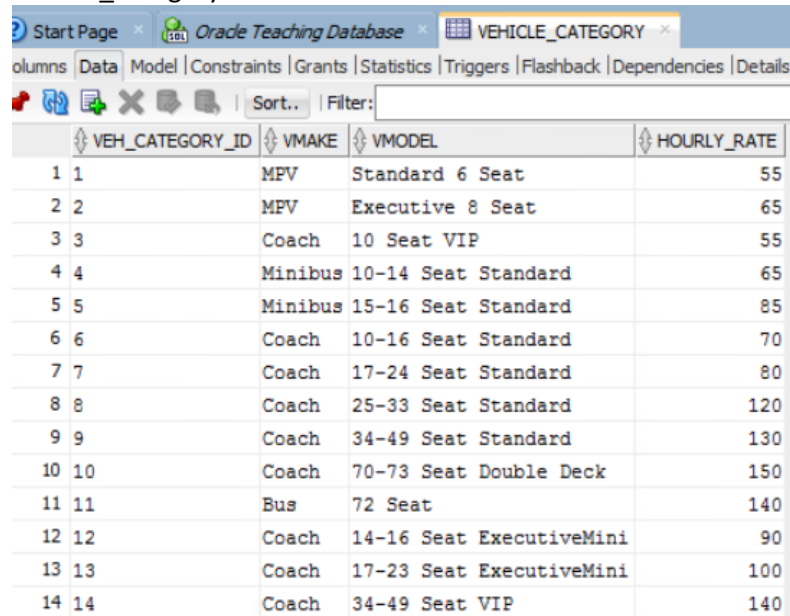
Task completed in 0.03 seconds

```
Driver_ID varchar2(8) REFERENCES Driver(Driver_ID),
Registration_Number varchar2(8) REFERENCES Vehicle(Registration_Number))
Error report -
ORA-00910: specified length too long for its datatype
00910. 00000 - "specified length too long for its datatype"
*Cause:    for datatypes CHAR and RAW, the length specified was > 2000;
           otherwise, the length specified was > 4000.
*Action:   use a shorter length or switch to a datatype permitting a
           longer length such as a VARCHAR2, LONG CHAR, or LONG RAW

Table BOOKING created.
```

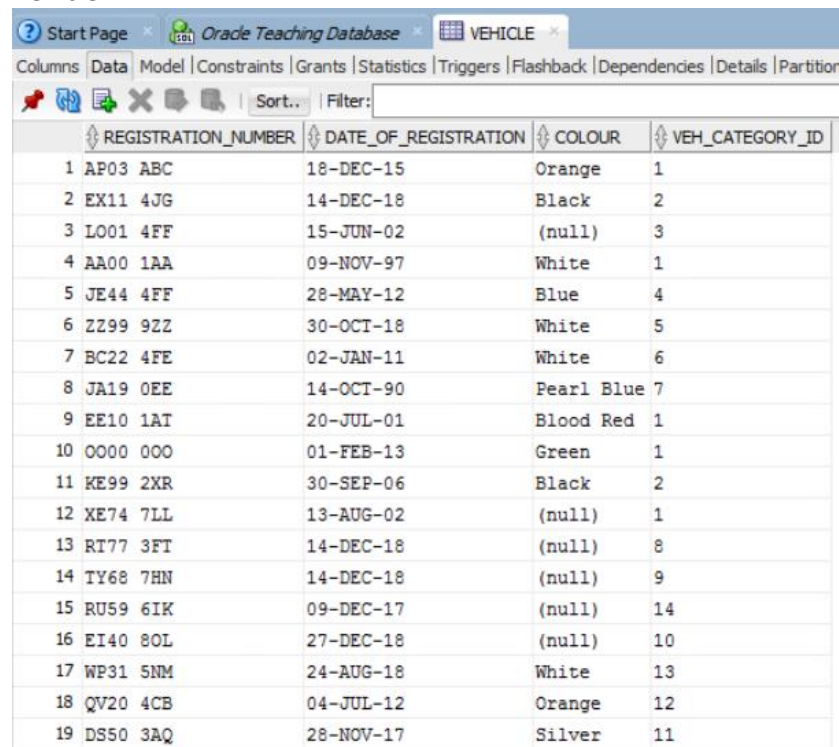
Task 4: Screenshot of populated data

Vehicle_Category



	VEH_CATEGORY_ID	VMAKE	VMODEL	HOURLY_RATE
1	1	MPV	Standard 6 Seat	55
2	2	MPV	Executive 8 Seat	65
3	3	Coach	10 Seat VIP	55
4	4	Minibus	10-14 Seat Standard	65
5	5	Minibus	15-16 Seat Standard	85
6	6	Coach	10-16 Seat Standard	70
7	7	Coach	17-24 Seat Standard	80
8	8	Coach	25-33 Seat Standard	120
9	9	Coach	34-49 Seat Standard	130
10	10	Coach	70-73 Seat Double Deck	150
11	11	Bus	72 Seat	140
12	12	Coach	14-16 Seat ExecutiveMini	90
13	13	Coach	17-23 Seat ExecutiveMini	100
14	14	Coach	34-49 Seat VIP	140

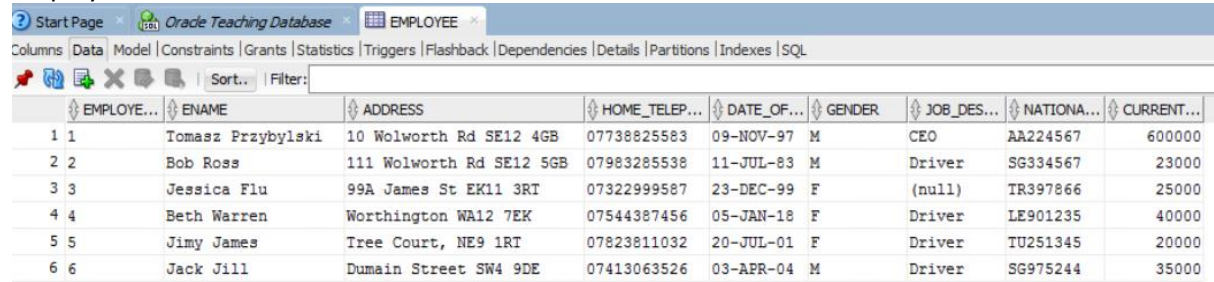
Vehicle



	REGISTRATION_NUMBER	DATE_OF_REGISTRATION	COLOUR	VEH_CATEGORY_ID
1	AP03 ABC	18-DEC-15	Orange	1
2	EX11 4JG	14-DEC-18	Black	2
3	LO01 4FF	15-JUN-02	(null)	3
4	AA00 1AA	09-NOV-97	White	1
5	JE44 4FF	28-MAY-12	Blue	4
6	ZZ99 9ZZ	30-OCT-18	White	5
7	BC22 4FE	02-JAN-11	White	6
8	JA19 OEE	14-OCT-90	Pearl Blue	7
9	EE10 1AT	20-JUL-01	Blood Red	1
10	OO00 000	01-FEB-13	Green	1
11	KE99 2XR	30-SEP-06	Black	2
12	XE74 7LL	13-AUG-02	(null)	1
13	RT77 3FT	14-DEC-18	(null)	8
14	TY68 7HN	14-DEC-18	(null)	9
15	RU59 6IK	09-DEC-17	(null)	14
16	EI40 8OL	27-DEC-18	(null)	10
17	WP31 5NM	24-AUG-18	White	13
18	QV20 4CB	04-JUL-12	Orange	12
19	DS50 3AQ	28-NOV-17	Silver	11

2018/19

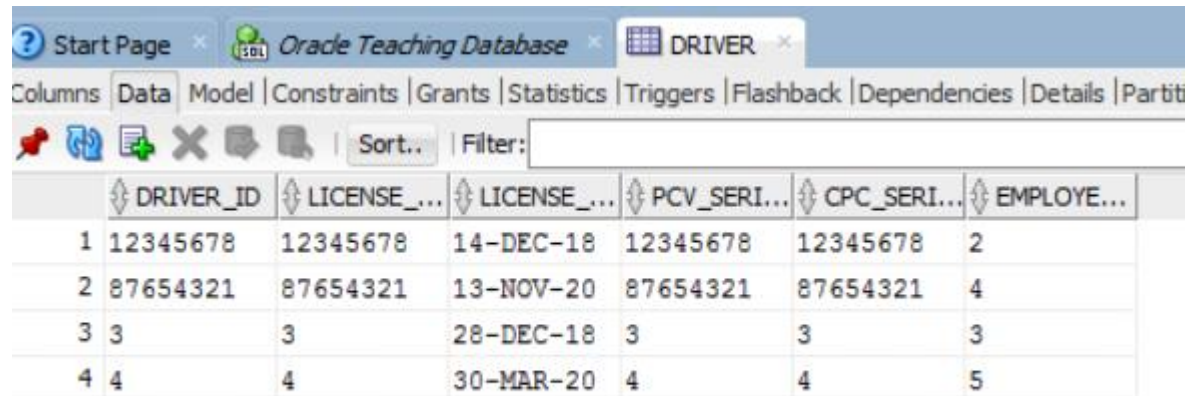
Employee



The screenshot shows the Oracle SQL Developer interface with the 'EMPLOYEE' table selected. The table has columns: EMPLOYEE_ID, ENAME, ADDRESS, HOME_PHONE, DATE_OF_BIRTH, GENDER, JOB_DESCRIPTION, NATIONALITY_ID, and CURRENT_SALARY. The data is as follows:

EMPLOYEE_ID	ENAME	ADDRESS	HOME_PHONE	DATE_OF_BIRTH	GENDER	JOB_DESCRIPTION	NATIONALITY_ID	CURRENT_SALARY
1	Tomasz Przybylski	10 Wolworth Rd SE12 4GB	07738825583	09-NOV-97	M	CEO	AA224567	600000
2	Bob Ross	111 Wolworth Rd SE12 5GB	07983285538	11-JUL-83	M	Driver	SG334567	23000
3	Jessica Flu	99A James St EK11 3RT	07322999587	23-DEC-99	F	(null)	TR397866	25000
4	Beth Warren	Worthington WA12 7EK	07544387456	05-JAN-18	F	Driver	LE901235	40000
5	Jimmy James	Tree Court, NE9 1RT	07823811032	20-JUL-01	F	Driver	TU251345	20000
6	Jack Jill	Dumain Street SW4 9DE	07413063526	03-APR-04	M	Driver	SG975244	35000

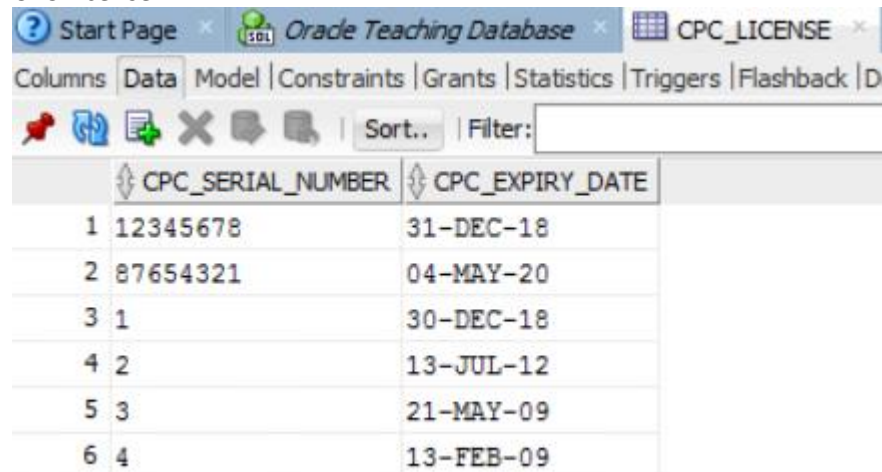
Driver



The screenshot shows the Oracle SQL Developer interface with the 'DRIVER' table selected. The table has columns: DRIVER_ID, LICENSE_NUMBER, LICENSE_EXPIRY_DATE, PCV_SERIAL_NUMBER, CPC_SERIAL_NUMBER, and EMPLOYEE_ID. The data is as follows:

DRIVER_ID	LICENSE_NUMBER	LICENSE_EXPIRY_DATE	PCV_SERIAL_NUMBER	CPC_SERIAL_NUMBER	EMPLOYEE_ID
1	12345678	14-DEC-18	12345678	12345678	2
2	87654321	13-NOV-20	87654321	87654321	4
3	3	28-DEC-18	3	3	3
4	4	30-MAR-20	4	4	5

CPC License

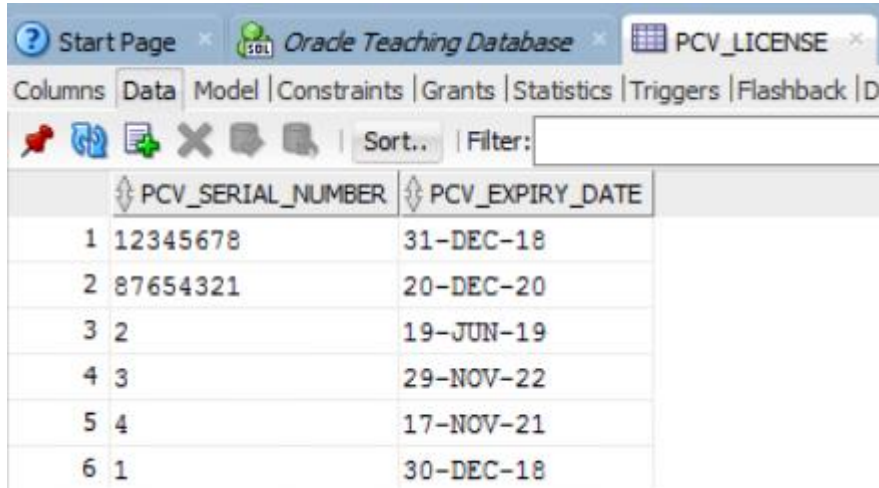


The screenshot shows the Oracle SQL Developer interface with the 'CPC_LICENSE' table selected. The table has columns: CPC_SERIAL_NUMBER and CPC_EXPIRY_DATE. The data is as follows:

CPC_SERIAL_NUMBER	CPC_EXPIRY_DATE
1	31-DEC-18
2	04-MAY-20
3	30-DEC-18
4	13-JUL-12
5	21-MAY-09
6	13-FEB-09

2018/19

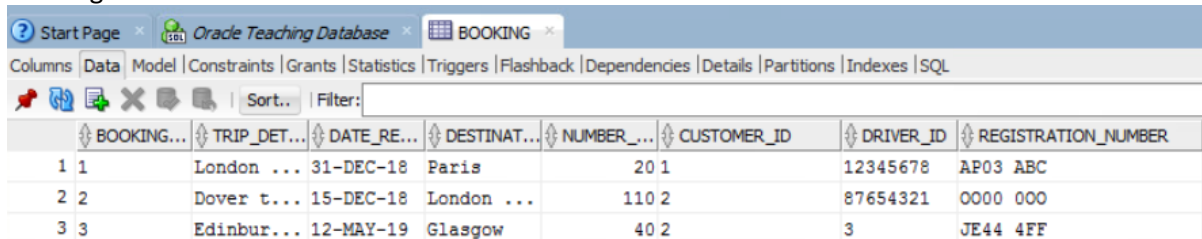
PCV License



The screenshot shows the Oracle SQL Developer interface with the 'PCV_LICENSE' table selected. The table has two columns: 'PCV_SERIAL_NUMBER' and 'PCV_EXPIRY_DATE'. The data is as follows:

	PCV_SERIAL_NUMBER	PCV_EXPIRY_DATE
1	12345678	31-DEC-18
2	87654321	20-DEC-20
3	2	19-JUN-19
4	3	29-NOV-22
5	4	17-NOV-21
6	1	30-DEC-18

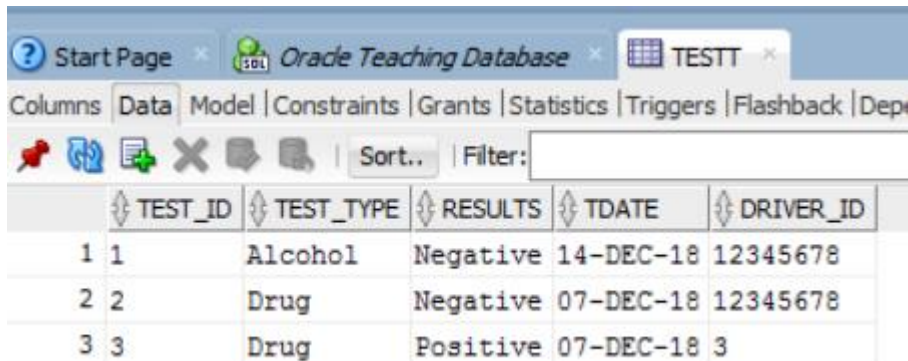
Booking



The screenshot shows the Oracle SQL Developer interface with the 'BOOKING' table selected. The table has eight columns: 'BOOKING...', 'TRIP_DET...', 'DATE_RE...', 'DESTINAT...', 'NUMBER_...', 'CUSTOMER_ID', 'DRIVER_ID', and 'REGISTRATION_NUMBER'. The data is as follows:

	BOOKING...	TRIP_DET...	DATE_RE...	DESTINAT...	NUMBER_...	CUSTOMER_ID	DRIVER_ID	REGISTRATION_NUMBER
1	1	London ...	31-DEC-18	Paris	20	1	12345678	AP03 ABC
2	2	Dover t...	15-DEC-18	London ...	110	2	87654321	0000 000
3	3	Edinbur...	12-MAY-19	Glasgow	40	2	3	JE44 4FF

TestT



The screenshot shows the Oracle SQL Developer interface with the 'TESTT' table selected. The table has six columns: 'TEST_ID', 'TEST_TYPE', 'RESULTS', 'TDATE', and 'DRIVER_ID'. The data is as follows:

	TEST_ID	TEST_TYPE	RESULTS	TDATE	DRIVER_ID
1	1	Alcohol	Negative	14-DEC-18	12345678
2	2	Drug	Negative	07-DEC-18	12345678
3	3	Drug	Positive	07-DEC-18	3

Task 5: Discussion of constraints used

Table Employee

Employee_ID uses a PRIMARY KEY constraint, this is to ensure each employee is uniquely identified

EName uses a NOT NULL constraint, this is to ensure that each employee has a name in the system

Address uses a NOT NULL constraint, this is to ensure that each employee has an address in the system

Home_Telephone_Number uses a NOT NULL constraint, this is to ensure that each employee has a telephone number in the system

Date_Of_Birth uses a NOT NULL constraint, this is to ensure that each employee has a DOB stored in the system

Gender uses a CHECK constraint, this is to ensure that each employee has a listed gender of "F" or "M", the two legal inputs

National_Insurance_Number uses a NOT NULL constraint, this is to ensure that each employee has a NI Number in the system

Current_Annual_Salary uses a CHECK constraint to ensure the salary is more than 0, and a NOT NULL constraint to ensure that each employee has a Current_Annual_Salary

Job_Description uses no constraints

Table Driver

Driver_ID uses a PRIMARY KEY constraint to ensure each Driver is uniquely identified

License_Number uses a UNIQUE constraint to ensure each Driver has a unique License_Number, and a NOT NULL constraint to ensure that each Driver has a License_Number

License_Expiry uses a NOT NULL constraint to ensure each Driver has a License_Expiry

PCV_Serial_Number uses a FOREIGN KEY constraint to ensure that a PCV_Serial_Number entered for a Driver exists in the PCV_License Table, and a NOT NULL constraint to ensure that each Driver has a PCV_Serial_Number

CPC_Serial_Number uses a FOREIGN KEY constraint to ensure that a CPC_Serial_Number entered for a Driver exists in the CPC_License Table, and a NOT NULL constraint to ensure that each Driver has a CPC_Serial_Number

Employee_ID uses a FOREIGN KEY constraint to ensure that a Employee_ID entered for a Driver exists in the Employee Table, and a NOT NULL constraint to ensure that each Driver has an Employee_ID

Table Test

Test_ID uses a PRIMARY KEY constraint to ensure each Test is uniquely identified

Test_Type uses a NOT NULL constraint to ensure that each Test has a Test_Type

Results uses a NOT NULL constraint to ensure that each Test has Results

TDate uses a DEFAULT constraint to ensure that TDate defaults to sysdate if nothing is entered in the table

Driver_ID uses a FOREIGN KEY constraint to ensure that a Driver_ID entered for a Test exists in the Driver Table, and a NOT NULL constraint to ensure each Test has a Driver_ID

Table PCV_License

PCV_Serial_Number uses a PRIMARY KEY constraint to ensure each PCV_License is uniquely identified

PCV_Expiry_Date uses a NOT NULL constraint to ensure each PCV_License has a PCV_Expiry_Date

Table CPC_License

CPC_Serial_Number uses a PRIMARY KEY constraint to ensure each CPC_License is uniquely identified

CPC_Expiry_Date uses a NOT NULL constraint to ensure each CPC_License has a CPC_Expiry_Date

Table Vehicle

Registration_Number uses a PRIMARY KEY constraint to ensure each Vehicle is uniquely identified

Date_of_Registration uses a NOT NULL constraint to ensure each Vehicle has a Date_of_Registration

Veh_Category_ID uses a FOREIGN KEY constraint to ensure that a Veh_Category_ID entered for a Vehicle exists in the Vehicle_Category Table, and a NOT NULL constraint to ensure each Vehicle has a Veh_Category_ID

Colour uses no constraints

Table Vehicle_Category

Veh_Category_ID uses a PRIMARY KEY constraint to ensure each Vehicle_Category is uniquely identified

VMake uses a NOT NULL constraint to ensure each Vehicle_Category has a VMake

VModel uses a NOT NULL constraint to ensure each Vehicle_Category has a VModel

Hourly_Rate uses a NOT NULL constraint to ensure each Vehicle_Category has an Hourly_Rate

Table Booking

Booking_ID uses a PRIMARY KEY constraint to ensure each Booking is uniquely identified

Date_Required uses a NOT NULL constraint to ensure each Booking has a Date_Required

Destination uses a NOT NULL constraint to ensure each Booking has a Destination

Number_of_Passangers uses a CHECK constraint to ensure the Number_of_Passangers is 0 or more, and a NOT NULL constraint to ensure each Booking has a Number_of_Passangers

Customer_ID uses a FOREIGN KEY constraint to ensure that a Customer_ID entered for a Booking exists in the Customer Table, and a NOT NULL constraint to ensure each Booking has a Customer_ID

Driver_ID uses a FOREIGN KEY constraint to ensure that a Driver_ID entered for a Booking exists in the Driver Table, and a NOT NULL constraint to ensure each Booking has a Driver_ID

Registration_Number uses a FOREIGN KEY constraint to ensure that a Registration_Number entered for a Booking exists in the Driver Table, and a NOT NULL constraint to ensure each Booking has a Registration_Number

Trip_Details uses no constraints

Table Customer

Customer_ID uses a PRIMARY KEY constraint to ensure each Customer is uniquely identified

Company_Name uses a NOT NULL constraint to ensure each Customer has a Company_Name

Contact_Name uses a NOT NULL constraint to ensure each Customer has a Contact_Name

Address uses a NOT NULL constraint to ensure each Customer has an Address

Email uses a NOT NULL constraint to ensure each Customer has an Email

Contact_Number uses a NOT NULL constraint to ensure each Customer has a Contact_Number

Task 6 (i): Query 1

Selecting everything from the Vehicle_Category table, ordering in descending order of Hourly_Rate.

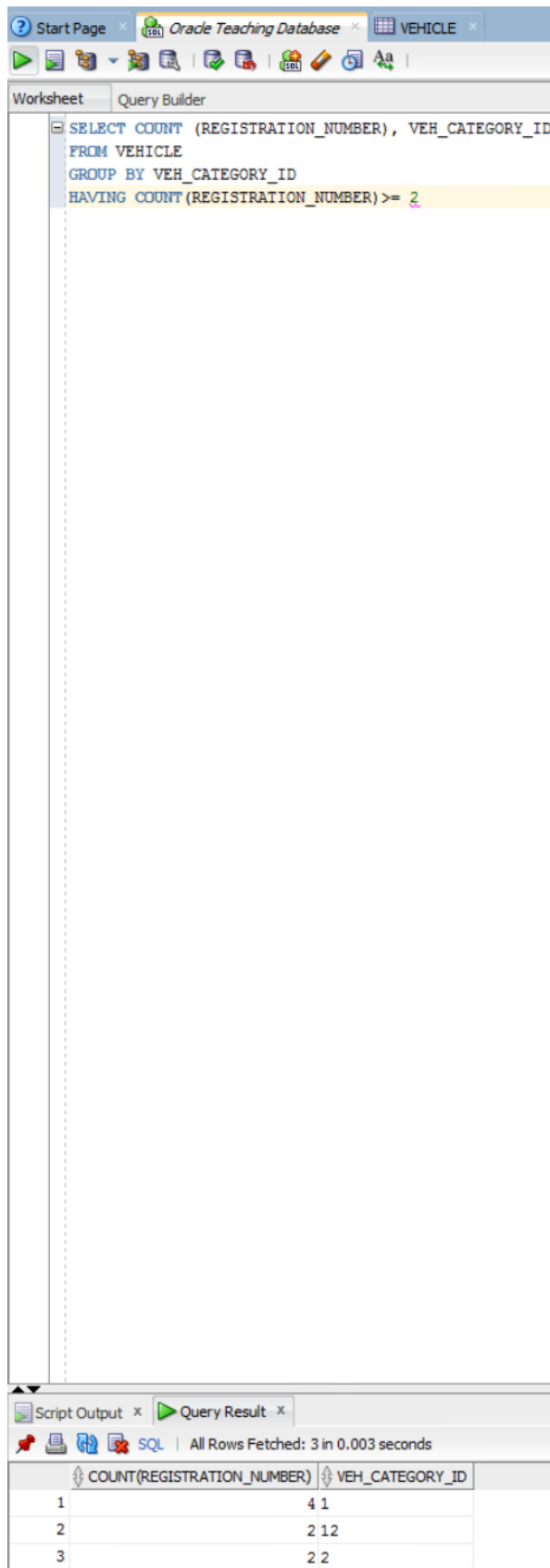
The screenshot shows the Oracle SQL Developer interface. The top pane displays the Query Builder with the following SQL query:

```
SELECT * FROM VEHICLE_CATEGORY
ORDER BY HOURLY_RATE DESC
```

The bottom pane shows the Query Result window with the following data:

VEH_CATEGORY_ID	VMAKE	VMODEL	HOURLY_RATE
1 10	Coach	70-73 Seat Double Deck	150
2 11	Bus	72 Seat	140
3 14	Coach	34-49 Seat VIP	140
4 9	Coach	34-49 Seat Standard	130
5 8	Coach	25-33 Seat Standard	120
6 13	Coach	17-23 Seat ExecutiveMini	100
7 12	Coach	14-16 Seat ExecutiveMini	90
8 5	Minibus	15-16 Seat Standard	85
9 7	Coach	17-24 Seat Standard	80
10 6	Coach	10-16 Seat Standard	70
11 4	Minibus	10-14 Seat Standard	65
12 2	MPV	Executive 8 Seat	65
13 3	Coach	10 Seat VIP	55
14 1	MPV	Standard 6 Seat	55

Task 6 (ii): Query 2



The screenshot shows the Oracle SQL Developer interface. The top pane displays a query in the Query Builder:

```
SELECT COUNT (REGISTRATION_NUMBER), VEH_CATEGORY_ID  
FROM VEHICLE  
GROUP BY VEH_CATEGORY_ID  
HAVING COUNT (REGISTRATION_NUMBER) >= 2
```

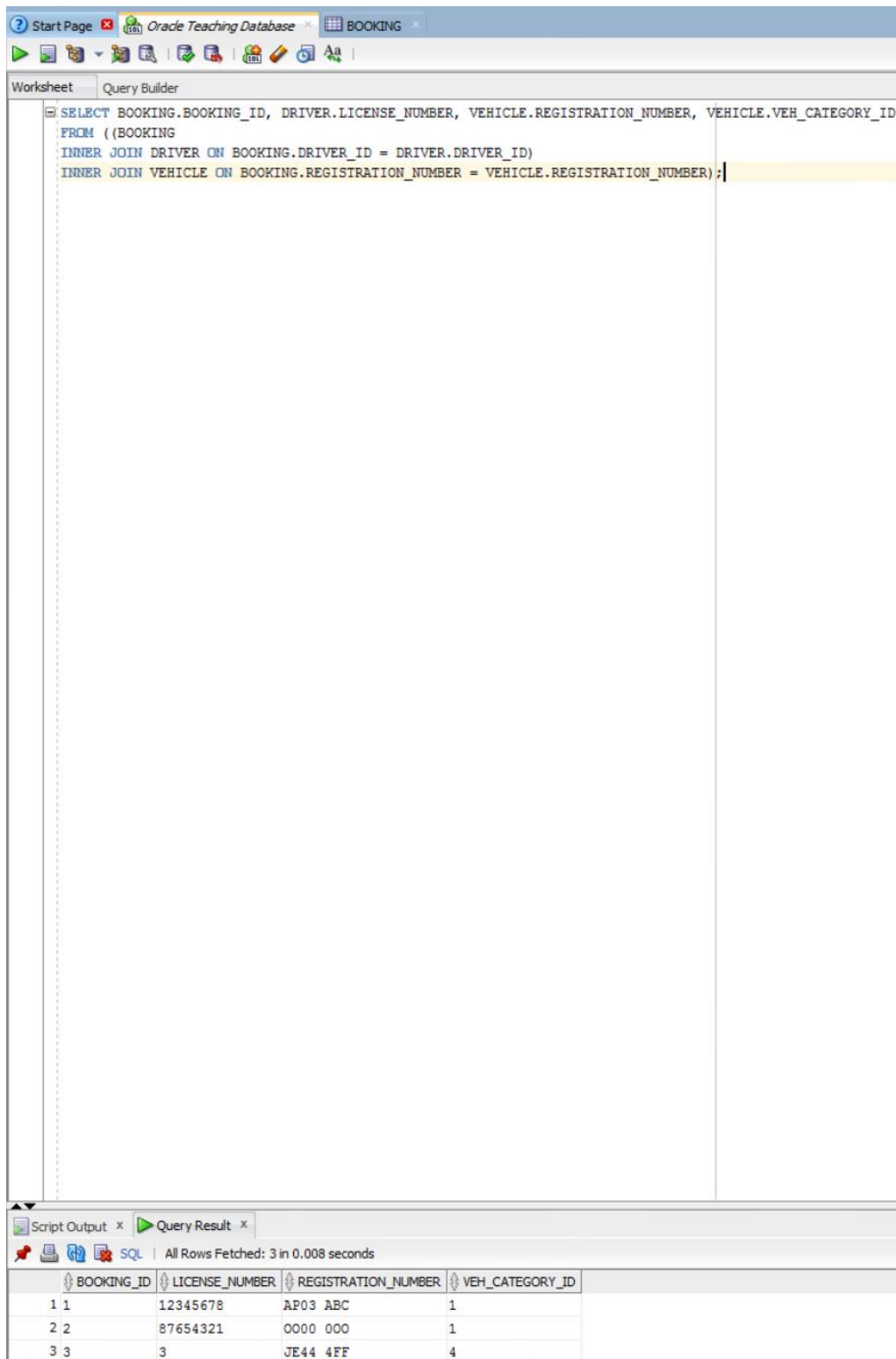
The bottom pane shows the Query Result with 3 rows fetched in 0.003 seconds. The results are as follows:

COUNT(REGISTRATION_NUMBER)	VEH_CATEGORY_ID
4	1
2	12
2	2

Select & count Registration_Number and Veh_Category_ID, and return any Veh_Category_ID that has 2 or more entries.

This returns the Veh_Category_IDs of 1, 2 and 12 as these are the Veh_Category_IDs that have 2 or more Registration_Numbers

Task 6 (iii): Query 3



The screenshot shows the Oracle SQL Developer interface. The top window is the 'Query Builder' for the 'BOOKING' table. The query being built is:

```
SELECT BOOKING.BOOKING_ID, DRIVER.LICENSE_NUMBER, VEHICLE.REGISTRATION_NUMBER, VEHICLE.VEH_CATEGORY_ID
FROM ((BOOKING
INNER JOIN DRIVER ON BOOKING.DRIVER_ID = DRIVER.DRIVER_ID)
INNER JOIN VEHICLE ON BOOKING.REGISTRATION_NUMBER = VEHICLE.REGISTRATION_NUMBER);
```

The bottom window is the 'Query Result' window, showing the results of the query. It indicates that all rows were fetched in 0.008 seconds. The results are displayed in a table with the following columns: BOOKING_ID, LICENSE_NUMBER, REGISTRATION_NUMBER, and VEH_CATEGORY_ID.

BOOKING_ID	LICENSE_NUMBER	REGISTRATION_NUMBER	VEH_CATEGORY_ID
1 1	12345678	AP03 ABC	1
2 2	87654321	0000 000	1
3 3	3	JE44 4FF	4

This query joins the Booking, Driver and Vehicle Tables and displays the Booking(Booking_ID), Driver(License_Number), Vehicle(Registration_Number) and Vehicle(Veh_Category_ID) wherever the Driver_ID match between the Booking and Driver tables, and wherever the Registration_Number matches between the Booking and Vehicle tables