

## Question 1

Create a database 'Hollywood', Table Movie- auto increment, pk.

```
datastudent@datastudent-VirtualBox: ~
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database Hollywood;
Query OK, 1 row affected (0.00 sec)

mysql> use Hollywood;
Database changed
mysql> create table Hollywood.Movie(mID INT Auto_increment primary key, title TEXT, year INT, director TEXT);
Query OK, 0 rows affected (0.04 sec)

mysql> Describe Movie;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| mID        | int(11)   | NO   | PRI | NULL    | auto_increment |
| title      | text      | YES  |     | NULL    |                |
| year       | int(11)   | YES  |     | NULL    |                |
| director   | text      | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

```
datastudent@datastudent-VirtualBox: ~
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 10
mysql> Delimiter $$
mysql> create trigger movie_year
-> before insert on Movie
-> for each row
-> begin
-> if new. year > 2016
-> then
-> signal sqlstate '02000' set message_text='Warning: year cannot be greater than 2016!';
-> end if;
-> end $$
Query OK, 0 rows affected (0.00 sec)

mysql> delimiter ;
mysql>
mysql>
mysql> ^C
mysql> insert into Movie(mID,title, year, director)
-> values('1','Gr', '2021', 'Farr');
ERROR 1643 (02000): Warning: year cannot be greater than 2016!
mysql>
```

5.  
The 'year' column in the Movie table should not be greater than 2016

MG

**Creating Tables Reviewer and Reting** –placing auto increment, pk's, default value of the 'ratingDate' column (as type timestamp and default current\_timestamp)

```
datastudent@datastudent-VirtualBox: ~  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> use Hollywood;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> create table Hollywood.Reviewer(rID INT AUTO_INCREMENT PRIMARY KEY, name  
TEXT0;  
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that  
corresponds to your MySQL server version for the right syntax to use near 'TEXT0  
' at line 1  
mysql> create table Hollywood.Reviewer(rID INT AUTO_INCREMENT PRIMARY KEY, name  
TEXT);  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> create table Hollywood.Reting(rID INT,mID INT, stars INT, ratingDate TIME  
STAMP default CURRENT_TIMESTAMP);  
Query OK, 0 rows affected (0.03 sec)  
  
mysql> DESCRIBE Reting;  
ERROR 1146 (42S02): Table 'Hollywood.Reting' doesn't exist  
mysql> describe Reting;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default      | Extra |  
+-----+-----+-----+-----+-----+-----+  
| rID        | int(11)   | YES  |     | NULL         |       |  
| mID        | int(11)   | YES  |     | NULL         |       |  
| stars      | int(11)   | YES  |     | NULL         |       |  
| ratingDate | timestamp | NO   |     | CURRENT_TIMESTAMP |       |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> alter table Reting add primary key (rID, mID);  
Query OK, 0 rows affected (0.05 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql> alter table Reting add constraint fk_rID foreign key (rID) references Rev
```

MG

## Adding Constrains FK's

```
datastudent@datastudent-VirtualBox: ~  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> alter table Reting add primary key (rID, mID);  
Query OK, 0 rows affected (0.05 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql> alter table Reting add constraint fk_rID foreign key (rID) references Reviewer (rID);  
Query OK, 0 rows affected (0.05 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql> alter table Reting add constraint fk_mID foreign key (mID) references Movie (mID);  
Query OK, 0 rows affected (0.05 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql> describe Reting;  
+-----+-----+-----+-----+-----+  
| Field      | Type      | Null | Key | Default          | Extra |  
+-----+-----+-----+-----+-----+  
| rID        | int(11)   | NO   | PRI | NULL             |       |  
| mID        | int(11)   | NO   | PRI | NULL             |       |  
| stars      | int(11)   | YES  |     | NULL             |       |  
| ratingDate | timestamp | NO   |     | CURRENT_TIMESTAMP |       |  
+-----+-----+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> describe Reviewer;  
+-----+-----+-----+-----+-----+  
| Field | Type      | Null | Key | Default | Extra      |  
+-----+-----+-----+-----+-----+  
| rID   | int(11)   | NO   | PRI | NULL    | auto_increment |  
| name  | text      | YES  |     | NULL    |               |  
+-----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql> 
```



MG

## Inserting Values to Reviewer and Rating

```
datastudent@datastudent-VirtualBox: ~
mysql>
mysql>
mysql> insert into Reviewer(rID, name) VALUES('1', 'John');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT*FROM Reviewer;
+-----+-----+
| rID | name |
+-----+-----+
| 1 | John |
+-----+-----+
1 row in set (0.00 sec)

mysql> SELECT*FROM Rating;
Empty set (0.00 sec)

mysql> insert into Rating(stars) VALUES(5);
ERROR 1364 (HY000): Field 'rID' doesn't have a default value
mysql> ^C
mysql> insert into Rating(rID) ^C
mysql> ;
ERROR:
No query specified

mysql> insert into Rating(rID, stars) VALUES('1', '5');
ERROR 1364 (HY000): Field 'mID' doesn't have a default value
mysql> insert into Rating(rID, mID, stars) VALUES('1','1','5');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT*FROM Rating;
+-----+-----+-----+-----+
| rID | mID | stars | ratingDate |
+-----+-----+-----+-----+
| 1 | 1 | 5 | 2021-03-05 16:15:06 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> █
```

MG

Showing all tables:

```
datastudent@datastudent-VirtualBox: ~  
  
mysql> show create table Movie;  
+-----+-----+  
+-----+  
| Table | Create Table  
+-----+  
|  
+-----+  
| Movie | CREATE TABLE `Movie` (  
  `mID` int(11) NOT NULL AUTO_INCREMENT,  
  `title` text,  
  `year` int(11) DEFAULT NULL,  
  `director` text,  
  PRIMARY KEY (`mID`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1 |  
+-----+  
+-----+  
1 row in set (0.00 sec)  
  
mysql> █
```

```
datastudent@datastudent-VirtualBox: ~  
  
+-----+  
| Reting | CREATE TABLE `Reting` (  
  `rID` int(11) NOT NULL,  
  `mID` int(11) NOT NULL,  
  `stars` int(11) DEFAULT NULL,  
  `ratingDate` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,  
  PRIMARY KEY (`rID`,`mID`),  
  KEY `fk_mID` (`mID`),  
  CONSTRAINT `fk_mID` FOREIGN KEY (`mID`) REFERENCES `Movie` (`mID`),  
  CONSTRAINT `fk_rID` FOREIGN KEY (`rID`) REFERENCES `Reviewer` (`rID`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |  
+-----+  
+-----+  
1 row in set (0.00 sec)  
  
mysql> █
```

```
datastudent@datastudent-VirtualBox: ~
-----+
1 row in set (0.00 sec)

mysql> show create table Reviewer;
+-----+
| Table      | Create Table
+-----+
| Reviewer   | CREATE TABLE `Reviewer` (
  `rID` int(11) NOT NULL AUTO_INCREMENT,
  `name` text,
  PRIMARY KEY (`rID`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1 |
+-----+
1 row in set (0.00 sec)

mysql> █
```

## Question 2

Executing the script:

```
datastudent@datastudent-VirtualBox: ~
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> DROP DATABASE IF EXISTS cind110A2Script1;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> CREATE SCHEMA cind110A2Script1;
Query OK, 1 row affected (0.00 sec)

mysql> USE cind110A2Script1;
Database changed
mysql> CREATE TABLE hiking (
  -> trail CHAR (50),
  -> area CHAR (50),
  -> distance FLOAT,
  -> est_time FLOAT);
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_cind110A2Script1 |
+-----+
| hiking                      |
+-----+
1 row in set (0.00 sec)

mysql> SHOW COLUMNS FROM hiking;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| trail      | char(50)  | YES  |     | NULL    |       |
| area       | char(50)  | YES  |     | NULL    |       |
| distance   | float     | YES  |     | NULL    |       |
| est_time   | float     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> INSERT INTO hiking VALUES
```



```

datastudent@datastudent-VirtualBox: ~
mysql> INSERT INTO hiking VALUES
    -> ( 'Cedar Creek Falls', 'Upper San Diego',4.5,2.5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO hiking(trail, area) VALUES
    -> ( 'East Mesa Loop', 'Cuyamaca Mountains');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM hiking;
+-----+-----+-----+-----+
| trail          | area          | distance | est_time |
+-----+-----+-----+-----+
| Cedar Creek Falls | Upper San Diego | 4.5      | 2.5      |
| East Mesa Loop   | Cuyamaca Mountains | NULL     | NULL     |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SET SQL_SAFE_UPDATES = 0;
Query OK, 0 rows affected (0.00 sec)

mysql> UPDATE hiking
    -> SET distance = 10.5, est_time = 5.5
    -> WHERE trail = 'East Mesa Loop';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> USE cind110A2Script1;
Database changed
mysql> DELETE FROM hiking WHERE trail = 'Cedar Creek Falls';
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM hiking;
+-----+-----+-----+-----+
| trail          | area          | distance | est_time |
+-----+-----+-----+-----+
| East Mesa Loop | Cuyamaca Mountains | 10.5     | 5.5      |
+-----+-----+-----+-----+

```

1. Write the SQL statements to insert the values into the hiking table:



MG

2. Write the SQL statements to update the entry for Oak Canyon

```
datastudent@datastudent-VirtualBox: ~
ERROR 1265 (01000): Data truncated for column 'est_time' at row 2
mysql> insert into hiking(trail, area, distance, est_time) VALUES('East Mesa Loop', 'Cuyamaca Mountains', 10.5, 10.5), ('Oak Canyon', NULL, 3, NULL);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select*from hiking;
+-----+-----+-----+-----+
| trail          | area          | distance | est_time |
+-----+-----+-----+-----+
| East Mesa Loop | Cuyamaca Mountains | 10.5    | 5.5      |
| East Mesa Loop | Cuyamaca Mountains | 10.5    | 10.5     |
| Oak Canyon     | NULL          | 3       | NULL     |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> delete from hiking where est_time=5.5;
Query OK, 1 row affected (0.00 sec)

mysql> select*from hiking;
+-----+-----+-----+-----+
| trail          | area          | distance | est_time |
+-----+-----+-----+-----+
| East Mesa Loop | Cuyamaca Mountains | 10.5    | 10.5     |
| Oak Canyon     | NULL          | 3       | NULL     |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> mysql> update hiking set area='Mission Trails Regional Park', est_time=2
where trail='Oak Canyon';
ERROR 1054 (42S22): Unknown column '2ime' in 'field list'
mysql> update hiking SET area='Mission Trails Regional Park', est_time= 2, WHERE
trail='Oak Canyon';
ERROR 1054 (42S22): Unknown column 'WHEREtrail' in 'field list'
mysql> update hiking set area='Mission Trails Regional Park', est_time=2 where a
rea = NULL and east_time= NULL;
```

```
datastudent@datastudent-VirtualBox: ~
mysql> update hiking SET area = 'Mission Trails Regional Park' WHERE area is NULL;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from hiking;
+-----+-----+-----+-----+
| trail          | area                  | distance | est_time |
+-----+-----+-----+-----+
| East Mesa Loop | Cuyamaca Mountains   | 10.5     | 10.5     |
| Oak Canyon     | Mission Trails Regional Park | 3        | NULL     |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> update hiking SET est_time = 2 WHERE est_time is NULL;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select*from hiking;
+-----+-----+-----+-----+
| trail          | area                  | distance | est_time |
+-----+-----+-----+-----+
| East Mesa Loop | Cuyamaca Mountains   | 10.5     | 10.5     |
| Oak Canyon     | Mission Trails Regional Park | 3        | 2        |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

3. Write the SQL statement to delete trails with a distance greater than 5 miles.
4. Write the SQL statement to create a table called 'rating'.

```

datastudent@datastudent-VirtualBox: ~
2 rows in set (0.00 sec)

mysql> DELETE from hiking WHERE distance > 5;
Query OK, 1 row affected (0.01 sec)

mysql> select*from hiking;
+-----+-----+-----+-----+
| trail          | area                               | distance | est_time |
+-----+-----+-----+-----+
| Oak Canyon    | Mission Trails Regional Park      | 3        | 2        |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> create table cind110A2Script1.rating(trail CHAR(50), dificulty INT);
Query OK, 0 rows affected (0.03 sec)

mysql> describe hiking;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| trail      | char(50)  | YES  |     | NULL    |       |
| area       | char(50)  | YES  |     | NULL    |       |
| distance   | float     | YES  |     | NULL    |       |
| est_time   | float     | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql> describe rating;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| trail      | char(50)  | YES  |     | NULL    |       |
| dificulty  | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> █

```

5. Write the command to add another column to the hiking table called 'trailID' with Primary key constraint.



MG

```
mysql> alter table hiking add column trailID INT PRIMARY KEY;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> describe rating;
```

Field	Type	Null	Key	Default	Extra
trail	char(50)	YES		NULL	
difficulty	int(11)	YES		NULL	

6. Add another column called 'trailID' in the 'rating' table and adding Constrains.

```
mysql> describe hiking;
```

Field	Type	Null	Key	Default	Extra
trail	char(50)	YES		NULL	
area	char(50)	YES		NULL	
distance	float	YES		NULL	
est_time	float	YES		NULL	
trailID	int(11)	NO	PRI	NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> ALTER TABLE rating add column trailID;
```

```
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 1
```

```
mysql> ALTER TABLE rating add column trailID INT;
```

```
Query OK, 0 rows affected (0.03 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> ALTER TABLE rating add CONSTRAINT fk_trailID FOREIGN KEY (trailID) REFERENCES hiking(trailID);
```

```
Query OK, 0 rows affected (0.04 sec)
```

```
Records: 0 Duplicates: 0 Warnings: 0
```

7. What is the command to delete the rating table?  
(See at the bottom)

MG

```
mysql> ^C
mysql> describe rating;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| trail      | char(50)  | YES  |     | NULL    |       |
| difficulty | int(11)   | YES  |     | NULL    |       |
| trailID    | int(11)   | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> describe hiking;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| trail      | char(50)  | YES  |     | NULL    |       |
| area       | char(50)  | YES  |     | NULL    |       |
| distance   | float     | YES  |     | NULL    |       |
| est_time   | float     | YES  |     | NULL    |       |
| trailID    | int(11)   | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> DROP TABLE hiking;
```

### Question 3

(Please make word file bigger (+ ) to see better the screenshots from mysql work bench)

MG

**Creating database CustOrders and adding tables Customer, Salesman and Order entities, constrains, inserting data:**

The image displays two screenshots of a database management tool interface, likely SQL Enterprise Manager, showing the execution of SQL scripts to create a database and insert data.

**Top Screenshot: Creating Database and Tables**

The left pane shows the 'SCHEMAS' tree with 'CustOrders' expanded, showing tables 'customers', 'orders', and 'salesman'. The right pane shows the SQL script being executed:

```
66 #####
67 #Q3 Assignment 2
68 • create database CustOrders;
69 • use CustOrders;
70
71 • create table CustOrders.customers(cust_id int primary key ,
72   cust_name text, city text, grade int, salesman_id int);
73
74 • create table CustOrders.salesman(salesman_id int primary key,
75   sales_name text, city text, commission float);
76
77 • create table CustOrders.orders(order_id int primary key,
78   purch_amt float, order_date date, cust_id int, salesman_id int);
79
80 • describe salesman;
81
82 #adding constraints
83
84 • alter table orders add constraint fk_cust_id foreign key (cust_id) references
85   customers(cust_id);
86 • alter table orders add constraint fk_salesman_id foreign key (salesman_id) references
87   customers(salesman_id);
88 • alter table customers add constraint fk_salesman_id foreign key (salesman_id)
89   references salesman(salesman_id);
```

The bottom pane shows the 'Object Info' tab for the 'salesman' table, indicating the primary key 'salesman\_id' (int(11) PK).

**Bottom Screenshot: Inserting Data**

The left pane shows the 'SCHEMAS' tree with 'CustOrders' expanded, showing tables 'customers', 'orders', and 'salesman'. The right pane shows the SQL script being executed:

```
91 • insert into customers(cust_id, cust name, city, grade, salesman_id)
92   values('3002', 'Nick Remando', 'New York', '100', '5001'),
93   ('3005', 'Graham Zusi', 'Caloifornia', '200', '5002'),
94   ('3001', 'Brad Guzan', 'London', null, '5005'),
95   ('3004', 'Fabian Johns', 'Paris', '300', '5006'),
96   ('3007', 'Brad Davis', 'New York', '200', '5001'),
97   ('3009', 'Geoff Camero', 'Berlin', '100', '5003'),
98   ('3008', 'Julian Green', 'London', '300', '5002'),
99   ('3003', 'Jozy Altidor', 'Moscow', '200', '5007');
100
101 • insert into salesman(salesman_id, sales_name, city, commission)
102   values('5001', 'James Hoog', 'New York', '0.15'),
103   ('5002', 'Nail Knite', 'Paris', '0.13'),
104   ('5005', 'Pit Alex', 'London', '0.11'),
105   ('5006', 'Mc Lyon', 'Paris', '0.14'),
106   ('5003', 'Lauson Hen', 'NULL', '0.12'),
107   ('5007', 'Paul Adam', 'Rome', '0.13');
108
109 • insert into orders(order_id, purch_amt, order_date, cust_id, salesman_id)
110   values('70001', '150.5', '2012-10-05', '3005', '5002'),
111   ('70009', '270.65', '2012-09-10', '3001', '5005'),
112   ('70002', '65.26', '2012-10-05', '3002', '5001'),
113   ('70004', '110.5', '2012-08-17', '3009', '5003'),
114   ('70007', '948.5', '2012-09-10', '3005', '5002'),
115   ('70005', '2400.6', '2012-07-27', '3007', '5001'),
116   ('70008', '5760', '2012-09-10', '3002', '5001'),
117   ('70010', '1983.43', '2012-10-10', '3004', '5006'),
118   ('70003', '2480.4', '2012-10-10', '3009', '5003');
```

The bottom pane shows the 'Object Info' tab for the 'salesman' table, indicating the primary key 'salesman\_id' (int(11) PK).



MG

The screenshot shows the SQL Developer interface for a local instance 3306. The left sidebar displays the 'SCHEMAS' tree with the 'salesman' table selected. The main query window contains the following SQL code:

```
123  
124 • describe customers;  
125 • describe salesman;  
126 • describe orders;  
127  
128  
129  
130  
131  
132  
133
```

The 'Result Grid' shows the structure of the 'salesman' table:

#	Field	Type	Null	Key	Default	Extra
1	cust_id	int(11)	NO	PRI		
2	cust_name	text	YES			
3	city	text	YES			
4	grade	int(11)	YES			
5	salesman_id	int(11)	YES	MUL		

The bottom status bar indicates 'Table: salesman', 'Columns: salesman\_id int(11) PK', and 'Query Completed'.

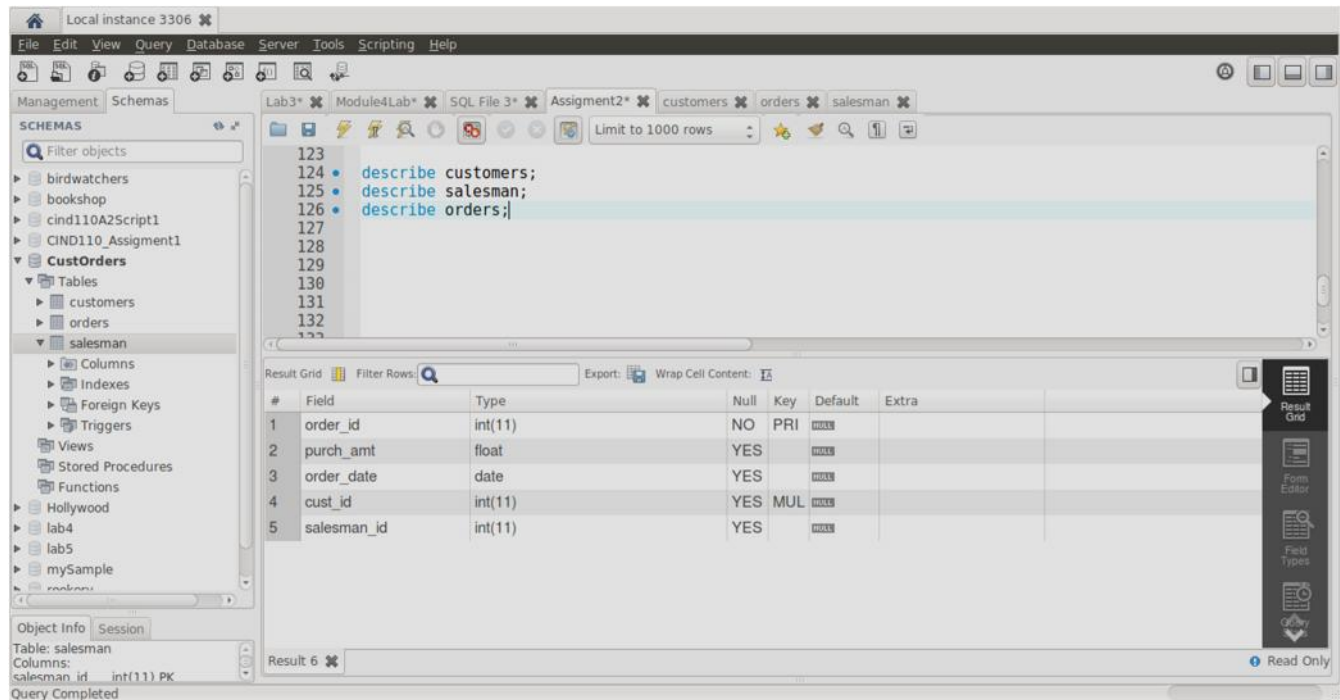
The screenshot shows the SQL Developer interface for a local instance 3306. The left sidebar displays the 'SCHEMAS' tree with the 'customers' table selected. The main query window contains the following SQL code:

```
123  
124 • describe customers;  
125 • describe salesman;  
126 • describe orders;  
127  
128  
129  
130  
131  
132  
133
```

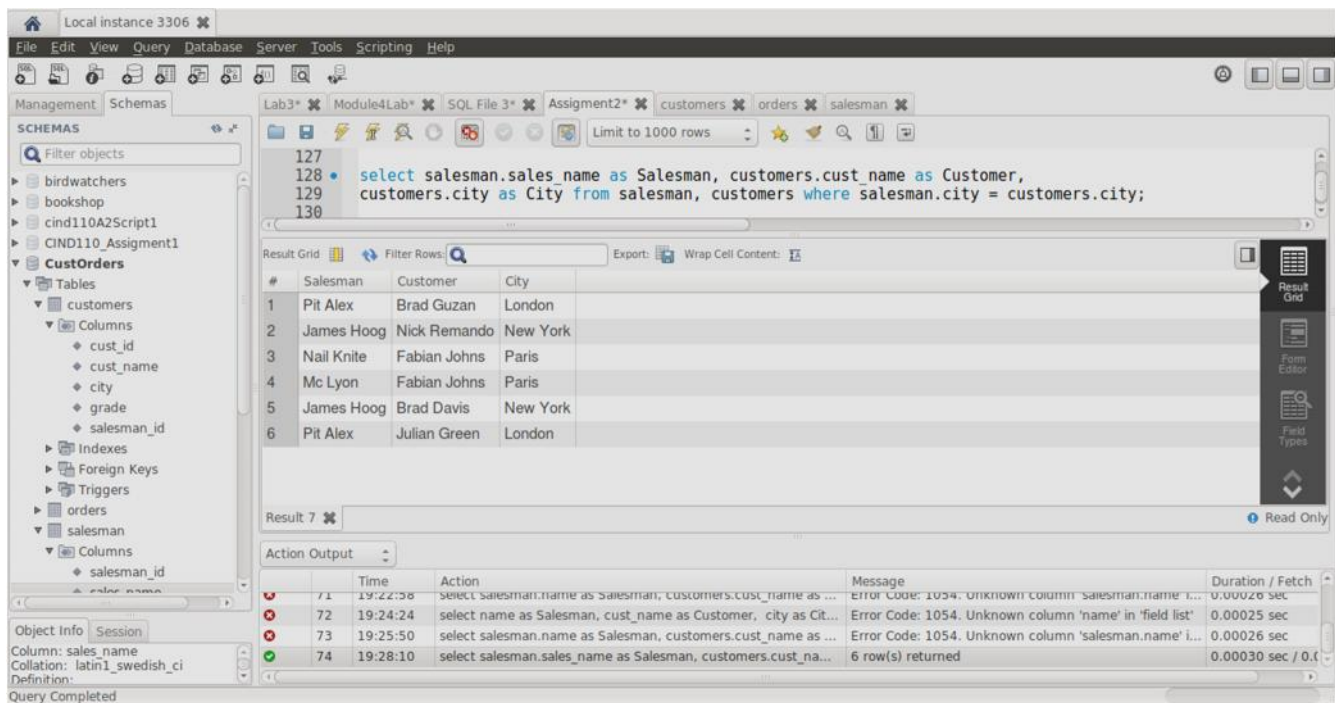
The 'Result Grid' shows the structure of the 'customers' table:

#	Field	Type	Null	Key	Default	Extra
1	salesman_id	int(11)	NO	PRI		
2	sales_name	text	YES			
3	city	text	YES			
4	commission	float	YES			

The bottom status bar indicates 'Table: salesman', 'Columns: salesman\_id int(11) PK', and 'Query Completed'.



1. Write an SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belong to same city.



2. Write an SQL statement to make a list with order no, purchase amount, customer name and their

MG

cities for the orders where order amount is between 500 and 2000.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'customers' and 'orders' tables expanded. The central pane shows a SQL query: `select orders.order_id, orders.purch_amt, customers.cust_name, customers.city from orders, customers where customers.cust_id = orders.cust_id and orders.purch_amt between 500 and 2000;`. The 'Result Grid' shows two rows of data:

#	order_id	purch_amt	cust_name	city
1	70007	948.5	Graham Zusi	Caloformia
2	70010	1983.43	Fabian Johns	Paris

The bottom pane shows the 'Action Output' table with two rows indicating the execution of the query.

3. Write an SQL statement to find out which salesmen are working for which customer.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'CustOrders' table expanded. The central pane shows a SQL query: `select salesman.sales_name as salesmen, customers.cust_name as customer FROM salesman, customers where salesman.salesman_id = customers.salesman_id`. The 'Result Grid' shows eight rows of data:

#	salesmen	customer
1	James Hoog	Nick Remando
2	James Hoog	Brad Davis
3	Nail Knite	Graham Zusi
4	Nail Knite	Julian Green
5	Lauson Hen	Geoff Camero
6	Pit Alex	Brad Guzan
7	Mc Lyon	Fabian Johns
8	Paul Adam	Jozy Altidor

The bottom pane shows the 'Action Output' table with two rows indicating the execution of the query.

4. Write an SQL statement to find the list of customers who appointed a salesman for their jobs whose commission is more than 12%.



The screenshot shows the SQL Developer interface with a query executed. The query is:

```

140 select salesman.sales_name as salesmen, customers.cust_name as customer, salesman.commission
141 FROM salesman, customers
142 where salesman.salesman_id = customers.salesman_id AND salesman.commission > 0.12;
143

```

The result grid shows the following data:

#	salesmen	customer	commission
1	James Hoog	Nick Remando	0.15
2	James Hoog	Brad Davis	0.15
3	Nail Knite	Graham Zusi	0.13
4	Nail Knite	Julian Green	0.13
5	Mc Lyon	Fabian Johns	0.14
6	Paul Adam	Jozy Altidor	0.13

The Action Output pane shows the execution details:

#	Time	Action	Message	Duration / Fetch
5	11:47:10	select salesman.sales_name as salesmen, customers.cust_name...	Error Code: 1064. You have an error in your SQL synta...	0.00024 sec
6	11:47:27	select salesman.sales_name as salesmen, customers.cust_name...	Error Code: 1064. You have an error in your SQL synta...	0.00030 sec
7	11:49:26	select salesman.sales_name as salesmen, customers.cust_name...	6 row(s) returned	0.00042 sec / 0.000...

Query Completed

5. Write an SQL statement to find the list of customers who appointed a salesman for their jobs who does not live in same city where the customer lives, and gets a commission above 12%.

The screenshot shows the SQL Developer interface with a query executed. The query is:

```

143
144 select customers.cust_name as customer, customers.city, salesman.sales_name as salesmen,
145 salesman.commission, salesman.city
146 FROM salesman, customers
147 where salesman.salesman_id = customers.salesman_id AND salesman.city <> customers.city
148 and salesman.commission > 0.12;
149
150

```

The result grid shows the following data:

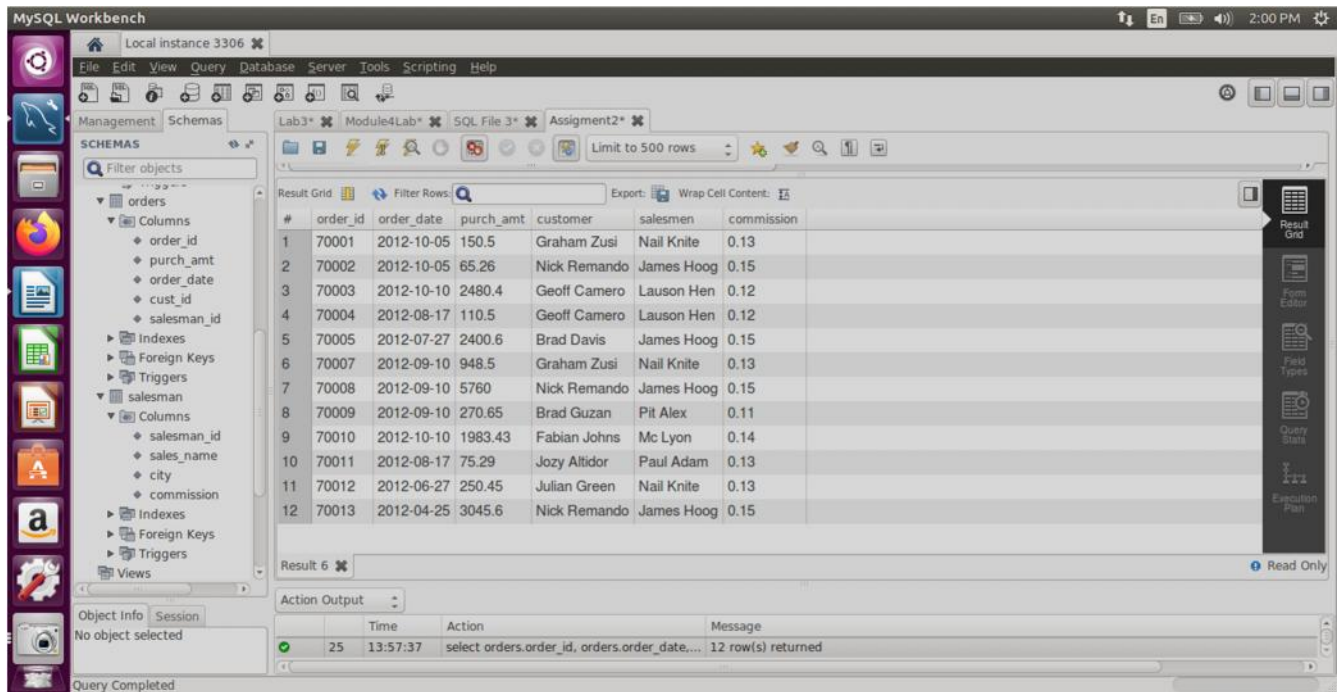
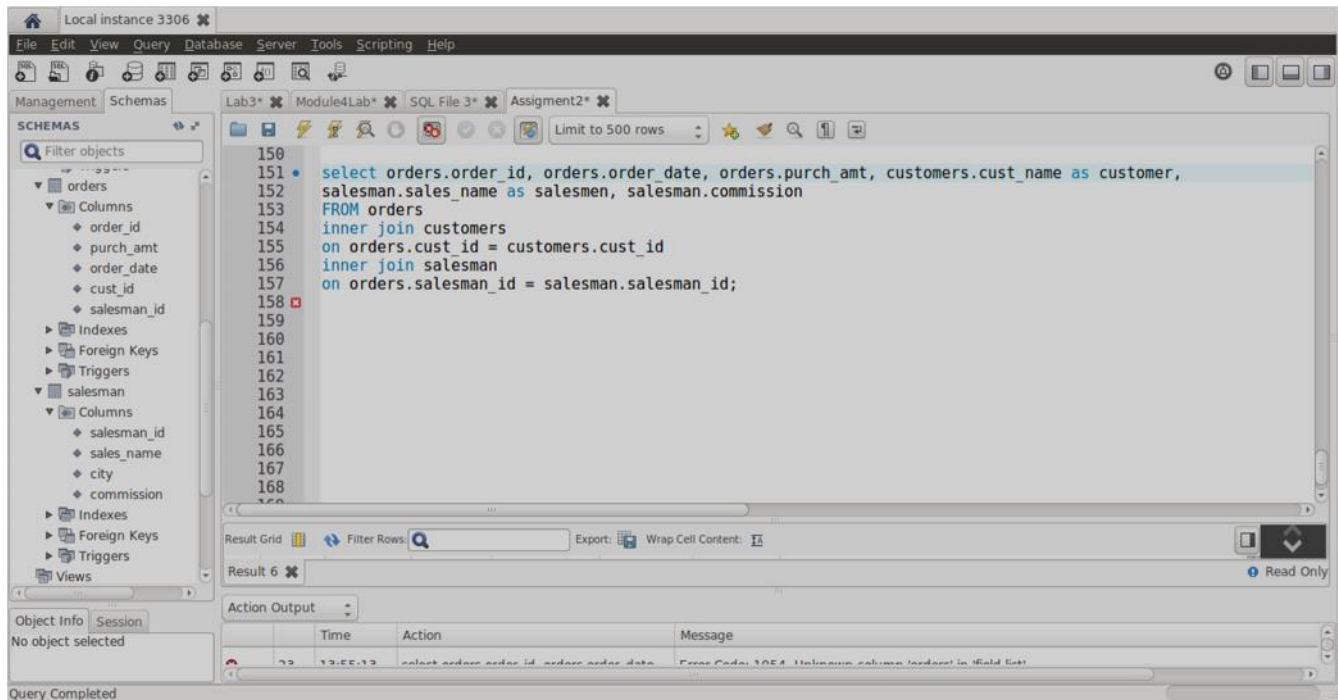
#	customer	city	salesmen	commission	city
1	Graham Zusi	Caloifornia	Nail Knite	0.13	Paris
2	Julian Green	London	Nail Knite	0.13	Paris
3	Jozy Altidor	Moscow	Paul Adam	0.13	Rome

The Action Output pane shows the execution details:

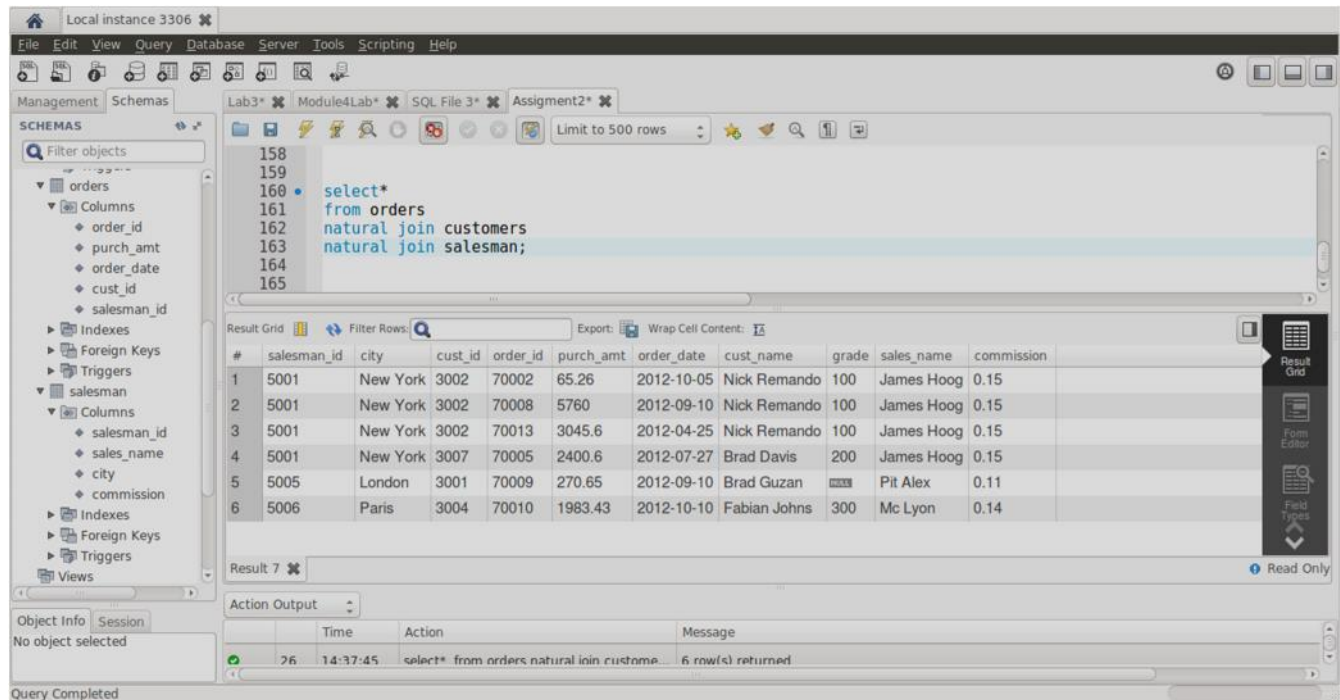
#	Time	Action	Message
10	12:13:03	select salesman.sales_name as salesmen,...	Error Code: 1004. You have an error in your SQL Syntax; check the manual that corresponds to your m...
11	12:14:01	use CustOrders	0 row(s) affected
12	12:19:46	select customers.cust_name as customer...	3 row(s) returned

Query Completed

6. Write an SQL statement to find the details of an order i. e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.



7. Write an SQL statement to make a join within the tables salesman, customer and orders such that the same column of each table will appear once and only the related rows will be returned.



## Question 4

Having that a Relation can have zero or more Foreign keys and each Foreign key can refer to different referenced Relations. Specify all possible Foreign keys for this schema

Please see my Schema as below with the all tables and relations, there are 4 FK's  
 Table Enroll will have 2FK's this is a table that connects tables Student and Course.  
 Table Book Adoption will have 2FK's this is a table that connects tables Course and Text.



