

---

*OMAR ISMAIL ABDJALEEL ALOMORY*

*Matric No: S63955*

*Date: 22/12/2022*

*Lab: MP3*

*Lecturer: Dr. ROSAIDA ROSLY*

*MUHAMMAD NASYRUL ADLY BIN MOHD AFANDY*

---

*DATABASE (CSF3123) K2*

## Activity 1

### Task 1

Create a tables based on ERD diagram and apply all constraints and referential integrity for related tables.

```
1   CREATE DATABASE ABC_College_Course_Registration;
2 • USE ABC_College_Course_Registration;
3
4   -- creating table for student =====
5 • Ⓜ CREATE TABLE lab4student(
6     stuid VARCHAR(10) PRIMARY KEY,
7     stuname VARCHAR(50),
8     stuaddr VARCHAR(70),
9     city VARCHAR(25),
10    state VARCHAR(25),
11    hphno VARCHAR(25),
12    stusstatus INT(11) not null
13 );
14
15   -- creating table for registration =====
16 • Ⓜ CREATE TABLE lab4registration(
17     regno VARCHAR(15) PRIMARY KEY,
18     stuid VARCHAR(10) not null,
19     regdate DATE not null,
20     pymtstatus CHAR(1) not null,
21     CONSTRAINT fk_lab4registration_stuid FOREIGN KEY(stuid) REFERENCES lab4student(stuid)
22 );
23
24   -- creating table for faculty table =====
25 • Ⓜ CREATE TABLE lab4faculty(
26     facultyid INT(3) PRIMARY KEY,
27     facultyname VARCHAR(50) NOT NULL,
28     office_no VARCHAR(15),
29     contactperson VARCHAR(50)
30 );
31
32   -- creating table for course =====
33 • Ⓜ CREATE TABLE lab4course(
34     crsid VARCHAR(15) PRIMARY KEY,
35     crsname VARCHAR(50),
36     crscredit INT(11) NOT NULL,
37     crslevel VARCHAR(15),
38     crsowner INT(3),
39     CONSTRAINT fk_lab4course_crsowner FOREIGN KEY(crsowner) REFERENCES lab4faculty(facultyid)
40 );
41
42   -- creating table for courseRegister =====
43 • Ⓜ CREATE TABLE lab4courseregister(
44     regno VARCHAR(15) not null,
45     crsid VARCHAR(15) not null,
46     crsstatus VARCHAR(12),
47     price DECIMAL(7,2),
48     CONSTRAINT fk_lab4courseregister_regno FOREIGN KEY (regno) REFERENCES lab4registration(regno),
49     CONSTRAINT fk_lab4courseregister_crsid FOREIGN KEY (crsid) REFERENCES lab4course(crsid)
50 );
```

## Task 2

Create a record for each table based on data provided in Table 1. (Mentioned in the lab module)

```
15      -- inserting values to student table
16 •  INSERT INTO lab4student (stuid, stuname, stuaddr, city, state, hpno, stustatus) VALUES
17      ('S12001', 'Ahmad Roslan', 'No.45, Presint 7', 'Putrajaya', 'Kuala Lumpur', '013-3841010',2),
18      ('S23008', 'Chan Wee Liew', 'DS-301, Desa Apartment', 'Kepong', 'Kuala Lumpur', '019-3884747',1),
19      ('S23002', 'Rosnah Rahman', 'Lot. 34, Kg. Seberang', 'Marang', 'Terengganu', '011-9957458',1),
20      ('S23037', 'Sri Shanti Devi', 'No. 7, Tmn Perwira', 'Kuantan', 'Pahang', '016-6685650',1),
21      ('S23025', 'Mohamad Ramli', 'No. 212, Jln Matang', 'Kuching', 'Sarawak', '012-8825698',1);
22      --
23      -- inserting values to registration table
24 •  INSERT INTO lab4registration(regno, stuid, regdate, pymtstatus) VALUES
25      ('R10011', 'S12001', '2015-09-22', 'Y'),
26      ('R10122', 'S23002', '2016-02-12', 'N'),
27      ('R10025', 'S23037', '2016-02-12', 'N'),
28      ('R10035', 'S23008', '2016-02-16', 'Y'),
29      ('R10049', 'S23025', '2016-02-26', 'N');
30      --
31      -- inserting values to faculty table
32 •  INSERT INTO lab4faculty(facultyid, facultyname, offic_no, contactperson) VALUES
33      (100, 'Faculty of Information Technology', '09-7751212', 'Mr. Sabri'),
34      (200, 'Faculty of Science', '09-7751010', 'Ms. Ramlah'),
35      (300, 'Faculty of Engineering', '09-7755678', 'Mr. Abd Ghani'),
36      (400, 'Faculty of Arts', '09-7751312', 'Mr. Eric'),
37      (500, 'Faculty of Medicine', '09-7758720', 'Ms. Hamidah');
38      --
39      -- inserting values to course
40 •  INSERT INTO lab4course(crsid, crsname, crscredit, crslevel, crsowner) VALUES
41      ('CS100', 'Introduction to Computer', 3, 'Degree', 100),
42      ('CS201', 'Basic Programming', 3, 'Degree', 100),
43      ('CS345', 'Introduction to Internet of Things', 4, 'Degree', 100),
44      ('EG522', 'Engineering Science', 4, 'Master', 300),
45      ('BG401', 'Bioinformatics', 3, 'Degree', 200);
46      --
47      -- inserting values to courseRegister
48 •  INSERT INTO lab4courseregister(regno, crsid, crsstatus, price) VALUES
49      ('R10011', 'CS100', 'Active', 450.00),
50      ('R10011', 'CS201', 'Active', 450.00),
51      ('R10011', 'CS345', 'Active', 450.00),
52      ('R10122', 'CS100', 'Active', 450.00),
53      ('R10122', 'CS201', 'Active', 450.00),
54      ('R10025', 'CS201', 'Active', 450.00),
55      ('R10025', 'BG401', 'Active', 450.00),
56      ('R10035', 'CS201', 'Active', 450.00),
57      ('R10035', 'CS345', 'Active', 450.00),
58      ('R10035', 'BG401', 'Active', 450.00),
59      ('R10049', 'EG522', 'Active', 600.00);
60      --
```

**Output:**

`SELECT * FROM lab4student;`

	stuid	stuname	stuaddr	city	state	hpno	stustatus
▶	S12001	Ahmad Roslan	No.45, Presint 7	Putrajaya	Kuala Lumpur	013-3841010	2
	S23002	Rosnah Rahman	Lot. 34, Kg. Seberang	Marang	Terengganu	011-9957458	1
	S23008	Chan Wee Liew	DS-301, Desa Apartment	Kepong	Kuala Lumpur	019-3884747	1
	S23025	Mohamad Ramli	No. 212, Jln Matang	Kuching	Sarawak	012-8825698	1
*	S23037	Sri Shanti Devi	No. 7, Tmn Perwira	Kuantan	Pahang	016-6685650	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

`SELECT * FROM lab4registration;`

	regno	stuid	regdate	pymtstatus
▶	R10011	S12001	2015-09-22	Y
	R10025	S23037	2016-02-12	N
	R10035	S23008	2016-02-16	Y
	R10049	S23025	2016-02-26	N
*	R10122	S23002	2016-02-12	N
*	NULL	NULL	NULL	NULL

`SELECT * FROM lab4faculty;`

	facultyid	facultyname	office_no	contactperson
▶	100	Faculty of Information Technology	09-7751212	Mr. Sabri
	200	Faculty of Science	09-7751010	Ms. Ramlah
	300	Faculty of Engineering	09-7755678	Mr. Abd Ghani
	400	Faculty of Arts	09-7751312	Mr. Eric
	500	Faculty of Medicine	09-7758720	Ms. Hamidah
*	NULL	NULL	NULL	NULL

`SELECT * FROM lab4course;`

	crsid	crsname	crscredit	crslevel	crsowner
▶	BG401	Bioinformatics	3	Degree	200
	CS100	Introduction to Computer	3	Degree	100
	CS201	Basic Programming	3	Degree	100
	CS345	Introduction to Internet of Things	4	Degree	100
*	EG522	Engineering Science	4	Master	300
*	NULL	NULL	NULL	NULL	NULL

`SELECT * FROM lab4courseregister;`

	regno	crsid	crsstatus	price
▶	R10011	CS100	Active	450.00
	R10011	CS201	Active	450.00
	R10011	CS345	Active	450.00
	R10122	CS100	Active	450.00
	R10122	CS201	Active	450.00
	R10025	CS201	Active	450.00
	R10025	BG401	Active	450.00
	R10035	CS201	Active	450.00
	R10035	CS345	Active	450.00
	R10035	BG401	Active	450.00
	R10049	EG522	Active	600.00

### Task 3

You are requested to retrieve registration information such registration number, student id and register date who still do not pay the fees and transform this records into a new table known as *pending\_payment*. Finally, retrieve all information from *pending\_payment* table.

First lets check the registration table and see who didn't pay yet:

```
SELECT * FROM lab4registration;
```

	regno	stuid	regdate	pymtstatus
▶	R10011	S12001	2015-09-22	Y
	R10025	S23037	2016-02-12	N
	R10035	S23008	2016-02-16	Y
	R10049	S23025	2016-02-26	N
*	R10122	S23002	2016-02-12	N
	NULL	NULL	NULL	NULL

These are the student  
who didn't pay yet

Now let's add them to another table:

```
94 • CREATE TABLE pending_payment
95     AS SELECT regno,stuid,regdate
96     FROM lab4registration
97     WHERE pymtstatus="N";
```

▼ abc\_college\_course\_registration

- ▼ Tables
  - lab4faculty
  - lab4courseregister
  - lab4course
  - lab4registration
  - lab4student
  - pending\_payment

We can see that the  
table is added

Once we created the table successfully, we can retrieve the information from it:

```
SELECT * FROM pending_payment;
```

	regno	stuid	regdate
▶	R10025	S23037	2016-02-12
	R10049	S23025	2016-02-26
*	R10122	S23002	2016-02-12

```
99 • SELECT * FROM lab4registration WHERE pymtstatus= 'N';
```

Result Grid | Filter Rows: | Edit: | Export/Import

	regno	stuid	regdate	pymtstatus
▶	R10025	S23037	2016-02-12	N
	R10049	S23025	2016-02-26	N
*	R10122	S23002	2016-02-12	N
	NULL	NULL	NULL	NULL

We can see that these  
values are the same

## Activity 2

```
3      -- 1. Create an index for student id's attribute in registration table and define the name of index as reg_studid_ind.  
4 • CREATE INDEX reg_studid_ind ON lab4registration(stuid);  
5 • SHOW INDEXES FROM lab4registration;
```

Result Grid												
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_c
lab4registration	0	PRIMARY	1	regno	A	0	NULL	NULL	BTREE			
lab4registration	1	reg_studid_ind	1	stuid	A	5	NULL	NULL	BTREE			

### Action output

```
✓ 107 23:36:49 CREATE INDEX reg_studid_ind ON lab4registration(stuid) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.078 sec
```

```
6      --  
7      -- 2. Create an index for attribute that representing the unique combination of student name and state in student  
8      -- table and define the index name as student_ind.  
9 • CREATE UNIQUE INDEX student_ind ON lab4student(stuname,state);  
10 • SHOW INDEXES FROM lab4student;
```

Result Grid														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expressio
lab4student	0	PRIMARY	1	stuid	A	0	NULL	NULL	BTREE			YES	NULL	
lab4student	0	student_ind	1	stuname	A	5	NULL	NULL	YES	BTREE		YES	NULL	
lab4student	0	student_ind	2	state	A	5	NULL	NULL	YES	BTREE		YES	NULL	

### Action output

```
✓ 111 23:53:16 CREATE UNIQUE INDEX student_ind ON lab4student(stuname,state) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.188 sec
```

```
11      --  
12      -- 3. Create an index for course name's attribute in course table and define the index name as crs_crsname_ind.  
13 • CREATE INDEX crs_crsname_ind ON lab4course(crsname);  
14 • SHOW INDEXES FROM lab4course;
```

Result Grid														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_cor	Visible	Expressio
lab4course	0	PRIMARY	1	crsname	A	0	NULL	NULL	BTREE			YES	NULL	
lab4course	1	fk_lab4course_crsowner	1	crsowner	A	0	NULL	NULL	YES	BTREE		YES	NULL	
lab4course	1	crs_crsname_ind	1	crsname	A	5	NULL	NULL	YES	BTREE		YES	NULL	

### Action output

```
✓ 115 23:57:01 CREATE INDEX crs_crsname_ind ON lab4course(crsname) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.063 sec
```

```
16      -- 4. Based on the latest analysis, DBA found that index crs_crsname_ind do not required since its would not affect  
17      -- the performance problem for the system. You are required to delete crs_crsname_ind from the system.  
18 • ALTER TABLE lab4course DROP INDEX crs_crsname_ind;  
19 • SHOW INDEXES FROM lab4course;
```

Result Grid														
Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_cor	Visible	Expressio
lab4course	0	PRIMARY	1	crsname	A	0	NULL	NULL	BTREE			YES	NULL	
lab4course	1	fk_lab4course_crsowner	1	crsowner	A	0	NULL	NULL	YES	BTREE		YES	NULL	

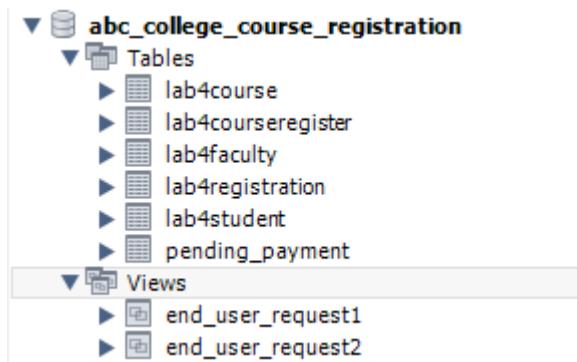
### Action output

```
✓ 119 00:02:34 ALTER TABLE lab4course DROP INDEX crs_crsname_ind 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.016 sec
```

### Activity 3

```
1  -- CREATING VIEWS FOR END-USERS --
2  ==
3  -- There is a two (2) requests from one of the end user to view the students who register the course
4  -- and to view student information. Based on this request, the management of ABC College will only
5  -- allow the end user to view information such as registration number, course id and course status
6  -- from the system. For student information, the end user is applicable to view only student
7  -- from Kuala Lumpur. As a Database Administrator (DBA) for ABC College, you are requested
8  -- to complete the request. Subsequently, verify these requests are created accordingly.
9
10 • USE abc_college_course_registration;
11 • CREATE VIEW end_user_request1 AS
12   SELECT
13     regno AS 'Registration No',
14     crsId AS 'Course ID',
15     crsstatus AS 'Course Status'
16   FROM lab4courseregisters;
17
18 • CREATE VIEW end_user_request2 AS
19   SELECT * FROM lab4student WHERE state = 'Kuala Lumpur';
```

Output			
#	Time	Action	Message
11	01:28:38	CREATE VIEW end_user_request2 AS SELECT * FROM lab4student WHERE state = 'Kuala Lumpur';	0 row(s) affected
12	01:28:43	CREATE VIEW end_user_request1 AS SELECT regno AS 'Registration No', crsId AS 'Course ID', crsstatus AS 'Course Status' FROM lab4courseregisters;	0 row(s) affected
11	01:28:38	CREATE VIEW end_user_request2 AS SELECT * FROM lab4student WHERE state = 'Kuala Lumpur';	0 row(s) affected
12	01:28:43	CREATE VIEW end_user_request1 AS SELECT regno AS 'Registration No', crsId AS 'Course ID', crsstatus AS 'Course Status' FROM lab4courseregisters;	0 row(s) affected



## Activity 4

```
1 -- Activity 4 -----
2
3 -- ABC College receive a new staff that will work at the ABC College Corporate Affairs department.
4 -- He will responsible to analyse the student intake and make a planning for next College
5 -- marketing program. As a DBA, you are requested to create a new database user as user01 and
6 -- password as user123 for this new staff and permit the user to view the information from the
7 -- system without doing any modification and creating any records. After completing these
8 -- requests, verify that user01 is only applicable to view data only.
9
10 • CREATE USER 'user01'@'localhost' IDENTIFIED BY 'user123';
11 • GRANT SELECT ON abc_college_course_registration.* TO 'user01'@'localhost';
```

User01 cannot do any modification to the data

```
13 • INSERT INTO lab4student VALUES
14   ('s52485','UMT','B2-201-5-1','Kuala Nerus','Terengganu','024-1241010',1);
15
16 • UPDATE lab4student SET city = 'Terengganu' WHERE stuid='S23037';
17 • DELETE FROM lab4student WHERE stuid='S23037';
```

Output:

#	Time	Action	Message
1	23:40:54	insert into lab4student values ('s52485','UMT','B2-201-5-1','Kuala Nerus','Terengganu','024-1241010',1)	Error Code: 1142. INSERT command denied to user 'user01'@'localhost' for table 'lab4student'
2	23:46:39	UPDATE lab4student SET city = 'Terengganu' WHERE stuid='S23037'	Error Code: 1142. UPDATE command denied to user 'user01'@'localhost' for table 'lab4student'
3	23:48:12	DELETE FROM lab4student WHERE stuid='S23037'	Error Code: 1142. DELETE command denied to user 'user01'@'localhost' for table 'lab4student'

User01 can only view the data.

```
19 • SELECT * FROM lab4student;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

stuid	stuname	stuaddr	city	state	hpno	stustatus
S12001	Ahmad Roslan	No.45, Presint 7	Putrajaya	Kuala Lumpur	013-3841010	2
S23002	Rosnah Rahman	Lot. 34, Kg. Seberang	Marang	Terengganu	011-9957458	1
S23008	Chan Wee Liew	DS-301, Desa Apartment	Kepong	Kuala Lumpur	019-3884747	1
S23025	Mohamed Ramli	No. 212, Jln Matang	Kuching	Sarawak	012-8825698	1
S23037	Sri Shanti Devi	No. 7, Tmn Perwira	Kuantan	Pahang	016-6685650	1

lab4student 2 ×

Output:

#	Time	Action	Message
1	23:40:54	insert into lab4student values ('s52485','UMT','B2-201-5-1','Kuala Nerus','Terengganu','024-1241010',1)	Error Code: 1142. INSERT command denied to user 'user01'@'localhost' for table 'lab4student'
2	23:46:39	UPDATE lab4student SET city = 'Terengganu' WHERE stuid='S23037'	Error Code: 1142. UPDATE command denied to user 'user01'@'localhost' for table 'lab4student'
3	23:48:12	DELETE FROM lab4student WHERE stuid='S23037'	Error Code: 1142. DELETE command denied to user 'user01'@'localhost' for table 'lab4student'
4	23:49:23	SELECT * FROM lab4student LIMIT 0, 1000	5 row(s) returned

GRANTS FOR user01

```
22 • SHOW GRANTS FOR 'user01'@'localhost';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Grants for user01@localhost
GRANT USAGE ON *.* TO 'user01' @'localhost' GRANT SELECT ON `abc_college_course_regist...

## Activity 5

```
1 -- Activity 5 -----
2 -- Use SQL DML commands to perform the following tasks;
3
4 -- 1. Retrieve student id, student name, address, city for student come from Kuala Lumpur add no
5 -- longer active in the ABC College.
```

First lets see the all the information for the student table to see who's status is not active

6 • `SELECT * from lab4student;`

The screenshot shows the MySQL Workbench interface with a result grid titled "Result Grid". The grid displays student data with columns: stuid, stuname, stuaddr, city, state, hpho, and stustatus. One row for student S12001 is highlighted with an orange border. A tooltip on the right side of the screen states: "'Ahmed Roslan' is not active as his status is '2'".

stuid	stuname	stuaddr	city	state	hpho	stustatus
S12001	Ahmed Roslan	No.45, Presint 7	Putrajaya	Kuala Lumpur	013-3841010	2
S23002	Rosnah Rahman	Lot. 34, Kg. Seberang	Marang	Terengganu	011-9957458	1
S23008	Chan Wee Liew	DS-301, Desa Apartment	Kepong	Kuala Lumpur	019-3884747	1
S23025	Mohamad Ramli	No. 212, Jln Matang	Kuching	Sarawak	012-8825698	1
S23037	Sri Shanti Devi	No. 7, Tmn Perwira	Kuantan	Pahang	016-6685650	1
*	NULL	NULL	NULL	NULL	NULL	NULL

lab4student 3 x

Output

Action Output

#	Time	Action
1	00:44:45	SELECT *from lab4student LIMIT 0, 1000

Message

5 row(s) returned

## **Result**

7 • `SELECT stuid, stuname, stuaddr FROM lab4student WHERE state = "Kuala Lumpur" AND stustatus = 2;`

The screenshot shows the MySQL Workbench interface with a result grid titled "Result Grid". The grid displays student data with columns: stuid, stuname, and stuaddr. One row for student S12001 is highlighted with an orange border. A message at the bottom right indicates "1 row(s) returned".

stuid	stuname	stuaddr
S12001	Ahmed Roslan	No.45, Presint 7
*	NULL	NULL

Output

Action Output

#	Time	Action
1	00:50:45	SELECT stuid, stuname, stuaddr FROM lab4student WHERE state = "Kuala Lumpur" AND stustatus = 2 ... 1 row(s) returned

Message

```

8   --
9   -- 2. Retrieve faculty information except for faculty code 300 and 500.
10 •  SELECT * FROM lab4faculty;

```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

facultyid	facultyname	office_no	contactperson
100	Faculty of Information Technology	09-7751212	Mr. Sabri
200	Faculty of Science	09-7751010	Ms. Ramlah
300	Faculty of Engineering	09-7755678	Mr. Abd Ghani
400	Faculty of Arts	09-7751312	Mr. Eric
500	Faculty of Medicine	09-7758720	Ms. Hamidah
*	HULL	HULL	HULL

lab4faculty 8 ×

Output :

Action Output

#	Time	Action
1	00:56:15	SELECT * FROM lab4faculty LIMIT 0, 1000

Message  
5 row(s) returned

## Result

```
11 •  SELECT * FROM lab4faculty where facultyid NOT IN (300,500);
```

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

facultyid	facultyname	office_no	contactperson
100	Faculty of Information Technology	09-7751212	Mr. Sabri
200	Faculty of Science	09-7751010	Ms. Ramlah
400	Faculty of Arts	09-7751312	Mr. Eric
*	HULL	HULL	HULL

lab4faculty 9 ×

Output :

Action Output

#	Time	Action
1	01:02:30	SELECT * FROM lab4faculty where facultyid NOT IN (300,500) LIMIT 0, 1000

Message  
3 row(s) returned

Now for question 3 in activity 4 I didn't understand the question properly, does it require to retrieve or modify.

If modify then modify what?

If it retrieves then retrieve all the information or only the latest one as it mentioned in the question.

This is all the info that the course having 3 credit and its owned by Faculty of Information Technology:

```
12  -- -----
13  -- 3. Modify credit value for course that having 3 credit hours and the course is owned by Faculty
14  -- of Information Technology. Retrieve the latest record.
15 •  SELECT c.* , f.facultynname FROM lab4course AS c INNER JOIN lab4faculty AS f ON f.facultyid = c.crsowner
16  WHERE c.crscredit = 3 AND f.facultynname = 'Faculty of Information Technology' ;-- LIMIT 1;
```

Result Grid					
crsid	crsname	crscredit	crslevel	crsowner	facultynname
CS100	Introduction to Computer	3	Degree	100	Faculty of Information Technology
CS201	Basic Programming	3	Degree	100	Faculty of Information Technology

Result 17		
Output		
Action Output		
#	Time	Action
1	01:21:43	SELECT c.* , f.facultynname FROM lab4course AS c INNER JOIN lab4faculty AS f ON f.facultyid = c.crsowner WHERE c.crscredit = 3 AND f.facultynname = 'Faculty of Information Technology' LIMIT 1; 2 row(s) returned

This is all the info for all the info that the course having 3 credit and its owned by Faculty of Information Technology. Retrieve the latest record:

```
18 •  SELECT c.* , f.facultynname FROM lab4course AS c INNER JOIN lab4faculty AS f ON f.facultyid = c.crsowner
19  WHERE c.crscredit = 3 AND f.facultynname = 'Faculty of Information Technology' LIMIT 1;
```

Result Grid					
crsid	crsname	crscredit	crslevel	crsowner	facultynname
CS100	Introduction to Computer	3	Degree	100	Faculty of Information Technology

Result 19		
Output		
Action Output		
#	Time	Action
1	01:24:09	SELECT c.* , f.facultynname FROM lab4course AS c INNER JOIN lab4faculty AS f ON f.facultyid = c.crsowner WHERE c.crscredit = 3 AND f.facultynname = 'Faculty of Information Technology' LIMIT 1; 1 row(s) returned

## Activity 6

```
1 -- Activity 6 -----
2 -- Complete the following request by applying SQL DML commands;
3
4 -- 1. Display the student id, name for posses character 'Ro'.
5 • SELECT stuid ,stuname FROM lab4student WHERE stuname LIKE 'Ro%';
6
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

stuid	stuname
S23002	Rosnah Rahman
NULL	NULL

Output

Action Output

#	Time	Action	Message
1	01:33:24	SELECT stuid ,stuname FROM lab4student WHERE stuname LIKE 'Ro%' LIMIT 0, 1000	1 row(s) returned

```
6 --
7 -- 2. Retrieve course id, course name and credit hours for course that begin with word as
8 -- 'Introduction'.
9 • SELECT crsid, crsname, crscredit FROM lab4course WHERE crsname LIKE 'Introduction%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

crsid	crsname	crscredit
CS100	Introduction to Computer	3
CS345	Introduction to Internet of Things	4
NULL	NULL	NULL

Output

Action Output

#	Time	Action	Message
1	01:39:26	SELECT crsid, crsname, crscredit FROM lab4course WHERE crsname LIKE 'Introduction%' LIMIT 0, 1000	2 row(s) returned

```
10 --
11 -- 3. Retrieve registration number, student id and registration date for the fifth characters
12 -- for student id which begin with the character '2'.
13 • SELECT regno, stuid, regdate FROM lab4registration WHERE stuid LIKE '____2%';
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

regno	stuid	regdate
R10049	S23025	2016-02-26
NULL	NULL	NULL

lab4registration11 x

Output

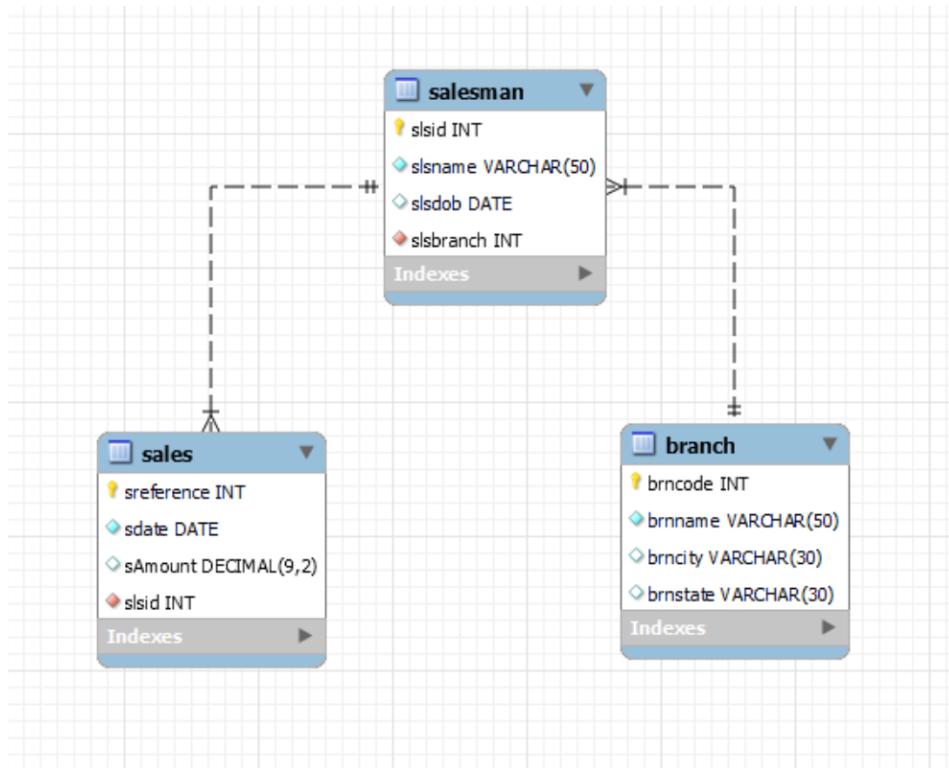
Action Output

#	Time	Action	Message
1	01:43:32	SELECT regno, stuid, regdate FROM lab4registration WHERE stuid LIKE '____2%' LIMIT 0, 1000	1 row(s) returned

## Lab Exercise

### Task 1

Draw a ERD diagram for this case study



### Task 2

Create a table for each entities by applying referential integrity and all the requirements stated in the a table. You should include SQL script and the output print screen of creating a table in your answer.

```
39 • Ⓜ CREATE TABLE Branch (
40     brncode INT (3) PRIMARY KEY ,
41     brnname VARCHAR(50) NOT NULL,
42     brncity VARCHAR(30),
43     brnstate VARCHAR(30)
44 );
9 • Ⓜ CREATE TABLE Salesman(
10     slsid INT(5) PRIMARY KEY AUTO_INCREMENT,
11     slsname VARCHAR(50) NOT NULL,
12     slsdob DATE DEFAULT (now()),
13     slsbranch INT(3) NOT NULL, -- FOREIGN KEY REFER TO BRANCH TABLE ON UPDATE CASCADE
14     CONSTRAINT fk_Salesman_slsbranch FOREIGN KEY (slsbranch) REFERENCES Branch (brncode)
15 );
16 • ALTER TABLE Salesman AUTO_INCREMENT =2000;
```

```

23      -- THIRD TO EXECUTE
24 • CREATE TABLE Sales (
25     sreference INT(8) PRIMARY KEY NOT NULL AUTO_INCREMENT,-- STARTS FROM 900000
26     sdate DATE NOT NULL,
27     sAmount DECIMAL(9,2),
28     slsid INT(5) NOT NULL,
29     CONSTRAINT fk_Sales_slsid FOREIGN KEY(slsid) REFERENCES Salesman (slsid)
30   );
31 • ALTER TABLE Sales AUTO_INCREMENT = 900000;

```

## Output

#	Time	Action	Message	Duration / Fetch
1	20:08:52	CREATE DATABASE abc_automart	1 row(s) affected	0.016 sec
2	20:08:53	USE abc_automart	0 row(s) affected	0.000 sec
3	20:09:08	CREATE TABLE Branch (brnco...	0 row(s) affected, 1 warning(s): 1681 Integer display width is deprecated and will be removed in a future rel...	0.016 sec
4	20:09:51	CREATE TABLE Salesman (slsid INT(5) PRIMARY KEY AUTO_INCREMENT, slname VARCHAR(50) N...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future rel...	0.016 sec
5	20:09:53	ALTER TABLE Salesman AUTO_INCREMENT =2000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
6	20:09:57	CREATE TABLE Sales (sreference INT(8) PRIMARY KEY NOT NULL AUTO_INCREMENT,-- STARTS ...	0 row(s) affected, 2 warning(s): 1681 Integer display width is deprecated and will be removed in a future rel...	0.031 sec
7	20:09:59	ALTER TABLE Sales AUTO_INCREMENT = 900000	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.016 sec

## Task 3

Insert the minimum TEN (10) records for each table you created in Task 2. You should refer to Task 4 before creating a record.

```

76 • insert into Branch (brnco...
```

- (110, 'Kuala Lumpur Main Branch', 'Kuala Lumpur', 'Kuala Lumpur'),
- (111, 'Kuala Lumpur East Branch', 'Kuala Lumpur', 'Kuala Lumpur'),
- (112, 'Kuala Lumpur West Branch', 'Kuala Lumpur', 'Kuala Lumpur'),
- (113, 'Johor Bahru Central Branch', 'Johor Bahru', 'Johor'),
- (116, 'Penang Island Branch', 'Penang', 'Penang'),
- (121, 'Kuching South Branch', 'Kuching', 'Sarawak'),
- (122, 'Ipoh Mian Branch', 'Ipoh', 'Perak'),
- (125, 'Kota Kinabalu Central Branch', 'Kota Kinabalu', 'Sabah'),
- (126, 'Kota Kinabalu North Branch', 'Kota Kinabalu', 'Sabah'),
- (127, 'Kota Kinabalu South Branch', 'Kota Kinabalu', 'Sabah'),
- (128, 'Shah Alam Central Branch', 'Shah Alam', 'Selangor'),
- (129, 'Shah Alam North Branch', 'Shah Alam', 'Selangor');

```
14 •    INSERT INTO Salesman ( slsname, slsdob, slsbranch) VALUES
15      ( 'Omar Alomory', '1998-01-01' , 110),
16      ( 'Hazim Mohammed', '1990-02-14' ,129),
17      ( 'MD Akash Ali', '1985-03-21' ,112),
18      ( 'Mohd Shafiq', '1999-02-22' ,113),
19      ( 'Gary Lim', '2000-03-03' ,113),
20      ( 'Danial Montaser', '1997-04-18' ,116),
21      ( 'Shahdad Ali', '1987-05-25' ,116),
22      ( 'Ali Amar', '1988-06-05' ,121),
23      ( 'Najmuldin Saleh', '1988-07-20' ,125),
24      ( 'Ajmal Mahdi', '1996-08-01' ,122),
25      ( 'Nancy Thompson', '1998-08-01' ,126),
26      ( 'Syafiah Adnan', '1989-08-01' ,127),
27      ( 'Shafiq VapeShop', '1993-08-01' ,128);
28


---


44 •    INSERT INTO Sales ( sdate, sAmount,slsid) VALUES
45      ( '2018-01-01' , 100000, 2001),
46      ( '2017-02-15' , 90000, 2009),
47      ( '2016-03-31' , 80000, 2010),
48      ( '2015-11-28' , 70000, 2012),
49      ( '2019-05-31' , 60000, 2013),
50      ( '2018-06-30' , 50000, 2001),
51      ( '2017-07-31' , 110000, 2002),
52      ( '2016-08-31' , 120000, 2003),
53      ( '2015-09-30' , 100000, 2004),
54      ( '2019-10-31' , 90000, 2005),
55      ( '2018-11-30' , 80000, 2006),
56      ( '2017-12-31' , 70000, 2007),
57      ( '2016-01-31' , 60000, 2008),
58      ( '2015-11-28' , 110000, 2009),
59      ( '2019-03-31' , 120000, 2009),
60      ( '2018-04-30' , 100000, 2010),
61      ( '2017-05-31' , 90000, 2011),
62      ( '2016-06-30' , 80000, 2012),
63      ( '2015-07-31' , 70000, 2013);
```

---

### Output

#	Time	Action	Message
✓	1 02:24:49	insert into Branch (brnCode, brnName, brnCity, brnState) VALUES (110, 'Kuala Lumpur Main Branch', 'Kuala Lumpur', 'Kuala Lumpur')	12 row(s) affected Records: 12 Duplicates: 0 Warnings: 0
✓	2 02:24:55	INSERT INTO Salesman (slsName, slsDob, slsBranch) VALUES ('Omar Alomory', '1998-01-01', 110), ('H...', 1998-01-01, 110)	13 row(s) affected Records: 13 Duplicates: 0 Warnings: 0
✓	3 02:24:57	INSERT INTO Sales (sdate, sAmount, slsID) VALUES ('2018-01-01', 100000, 2001), ('2017-02-15', 90000, 2001)	19 row(s) affected Records: 19 Duplicates: 0 Warnings: 0

## Task 4

Using SQL Data Definition Language (DDL) and Data Manipulation Language (DML) command, complete the following tasks;

```

94      --
95      -- Retrieve a list of sales reference, selling date and selling price for any sales created in the
96      -- system on 28 November 2015 and the selling price is greater than RM75K.
97
98 •   SELECT sreference, sdate , sAmount FROM Sales WHERE sdate = "2015-11-28" AND sAmount >75000;
--
```

	sreference	sdate	sAmount
▶	900014	2015-11-28	110000.00
*	NULL	NULL	NULL

brnCode	brnName	brnCity	brnState
110	Kuala Lumpur Main Branch	Kuala Lumpur	Kuala Lumpur
111	Kuala Lumpur East Branch	Kuala Lumpur	Kuala Lumpur
112	Kuala Lumpur West Branch	Kuala Lumpur	Kuala Lumpur

Action Output

#	Time	Action	Message
✓	1 02:24:49	insert into Branch (brnCode, brnName, brnCity, brnState) VALUES (110, 'Kuala Lumpur Main Branch', 'Kuala Lumpur', 'Kuala Lumpur')	12 row(s) affected Records: 12 Duplicates: 0 Warnings: 0
✓	2 02:24:55	INSERT INTO Salesman (slsName, slsDob, slsBranch) VALUES ('Omar Alomory', '1998-01-01', 110), ('H...', 1998-01-01, 110)	13 row(s) affected Records: 13 Duplicates: 0 Warnings: 0
✓	3 02:24:57	INSERT INTO Sales (sdate, sAmount, slsID) VALUES ('2018-01-01', 100000, 2001), ('2017-02-15', 90000, 2001)	19 row(s) affected Records: 19 Duplicates: 0 Warnings: 0
✓	4 02:27:34	SELECT sreference, sdate , sAmount FROM Sales WHERE sdate = "2015-11-28" AND sAmount >75000;	1 row(s) returned
✓	5 02:27:57	CREATE TABLE KLBranch AS SELECT * FROM Branch WHERE brnState = "Kuala Lumpur"	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0
✓	6 02:30:29	SELECT * FROM KLBranch LIMIT 0, 1000	3 row(s) returned

```

106      --
107      -- Create a new index known as salesman_slsname_ind to improve the performance when
108      -- retrieving specific salesman.
109 •   CREATE INDEX salesman_slsname_ind ON Salesman(slsName, slsDob);
110 •   SHOW INDEXES FROM Salesman;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
salesman	0	PRIMARY	1	slsID	A	13	NULL	NULL	BTREE				YES	NULL
salesman	1	fk_Salesman_slsbranch	1	slsBranch	A	11	NULL	NULL	BTREE				YES	NULL
salesman	1	salesman_slsname_ind	1	slsName	A	13	NULL	NULL	BTREE				YES	NULL
salesman	1	salesman_slsname_ind	2	slsDob	A	13	NULL	YES	BTREE				YES	NULL

Action Output

#	Time	Action	Message
✓	8 02:33:32	CREATE INDEX salesman_slsname_ind ON Salesman(slsName, slsDob)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
✓	9 02:34:02	SHOW INDEXES FROM Salesman	4 row(s) returned

sales  
salesman  
Views  
sales\_information  
Stored Procedures  
Functions  
abc\_college\_course\_registration  
sys

```

112  --
113  -- 4. ABC AutoMart Sdn Bhd top management wants to provide sales information to all salesman
114  -- across Malaysia without exposing the sales reference number and sales amount for each
115  -- sale. As a Database Administrator (DBA), you are requested to complete this task.
116 • CREATE VIEW Sales_Information AS SELECT sdate, slsid FROM Sales;
117 • SELECT * from Sales_information;

```

**Result Grid** | Filter Rows: Export: Wrap Cell Content:

	sdate	slsid
▶	2018-01-01	2001
	2017-02-15	2009
	2016-03-31	2010
	2015-11-28	2012
	2019-05-31	2013
	2018-06-30	2001
	2017-07-31	2002
	2016-08-31	2003
	2015-09-30	2004
	2019-10-31	2005
	2018-11-30	2006
	2017-12-31	2007

Sales\_information 15 ×

**Output** :

Action Output

#	Time	Action	Message
10	02:35:44	CREATE VIEW Sales_Information AS SELECT sdate, slsid FROM Sales	0 row(s) affected
11	02:36:07	SELECT * from Sales_information LIMIT 0, 1000	19 row(s) returned

Object Info Session

```

122  --
123  -- Display salesman id, name and their branch for the name contain 'Mohd'
124 • SELECT slsid, slsname, slsbranch FROM Salesman WHERE slsname LIKE '%Mohd%';
125  --

```

**Result Grid** | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	slsid	slsname	slsbranch
▶	2004	Mohd Shafiq	113
*	NULL	NULL	NULL

Salesman 16 ×

**Output** :

Action Output

#	Time	Action	Message
11	02:36:07	SELECT * from Sales_information LIMIT 0, 1000	19 row(s) returned
12	02:37:56	SELECT slsid, slsname, slsbranch FROM Salesman WHERE slsname LIKE '%Mohd%' LIMIT 0, 1000	1 row(s) returned

```

125      --
126      -- Display branch code, name and city for city which the first character for city begin with
127      -- alphabet 'K'.
128 •   SELECT brncode, brnname, brncity FROM Branch WHERE brncity LIKE "K%";
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	brncode	brnname	brncity
▶	110	Kuala Lumpur Main Branch	Kuala Lumpur
	111	Kuala Lumpur East Branch	Kuala Lumpur
	112	Kuala Lumpur West Branch	Kuala Lumpur
	121	Kuching South Branch	Kuching
	125	Kota Kinabalu Central Branch	Kota Kinabalu
	126	Kota Kinabalu North Branch	Kota Kinabalu
*	127	Kota Kinabalu South Branch	Kota Kinabalu
	NULL	NULL	NULL

Branch 17 ×

Output

Action Output

#	Time	Action	Message
✓	12 02:37:56	SELECT slsid, slsname, slsbranch FROM Salesman WHERE slsname LIKE "%Mohd%" LIMIT 0, 1000	1 row(s) returned
✓	13 02:38:37	SELECT bmcode, bmname, bmcity FROM Branch WHERE bmcity LIKE "K%" LIMIT 0, 1000	7 row(s) returned

```

129      --
130      -- Modify the salesman name to 'Fairuz Nasaruddin' for salesman id = 2007 and from branch
131      -- 112. It was Shahdad something
132 •   SELECT * FROM Salesman;
133 •   UPDATE Salesman SET slsname ="Fairuz Nasaruddin" WHERE slsid = 2007;
134      --
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

	slsid	slsname	slsdob	slsbranch
▶	2001	Omar Alomory	1998-01-01	110
	2002	Hazim Mohammed	1990-02-14	129
	2003	MD Akash Ali	1985-03-21	112
	2004	Mohd Shafiq	1999-02-22	113
	2005	Gary Lim	2000-03-03	113
	2006	Danial Montase	1997-04-18	116
	2007	Fairuz Nasaruddin	1987-05-25	116
	2008	Ali Amar	1988-06-05	121
	2009	Najmuldin Saleh	1988-07-20	125
	2010	Ajimal Mahdi	1996-08-01	122
	2011	Nancy Thompson	1998-08-01	126
	2012	Syafah Adnan	1989-08-01	127
	2013	Shafiq VapeShop	1993-08-01	128
*	NULL	NULL	NULL	NULL

Salesman 21 ×

Output

Action Output

#	Time	Action	Message
✓	18 02:43:53	UPDATE Salesman SET slsname ="Fairuz Nasaruddin" WHERE slsid = 2007	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
✓	19 02:43:59	SELECT * FROM Salesman LIMIT 0, 1000	13 row(s) returned

```

134  --
135  -- Create a user user04 with password 'u04@ser'. Grant the SELECT role to this user to view
136  -- sales table.
137 • CREATE USER 'user04'@'localhost' IDENTIFIED BY 'U04@ser';
138 • GRANT SELECT ON abc_automart.sales TO 'user04'@'localhost';
139  --

```

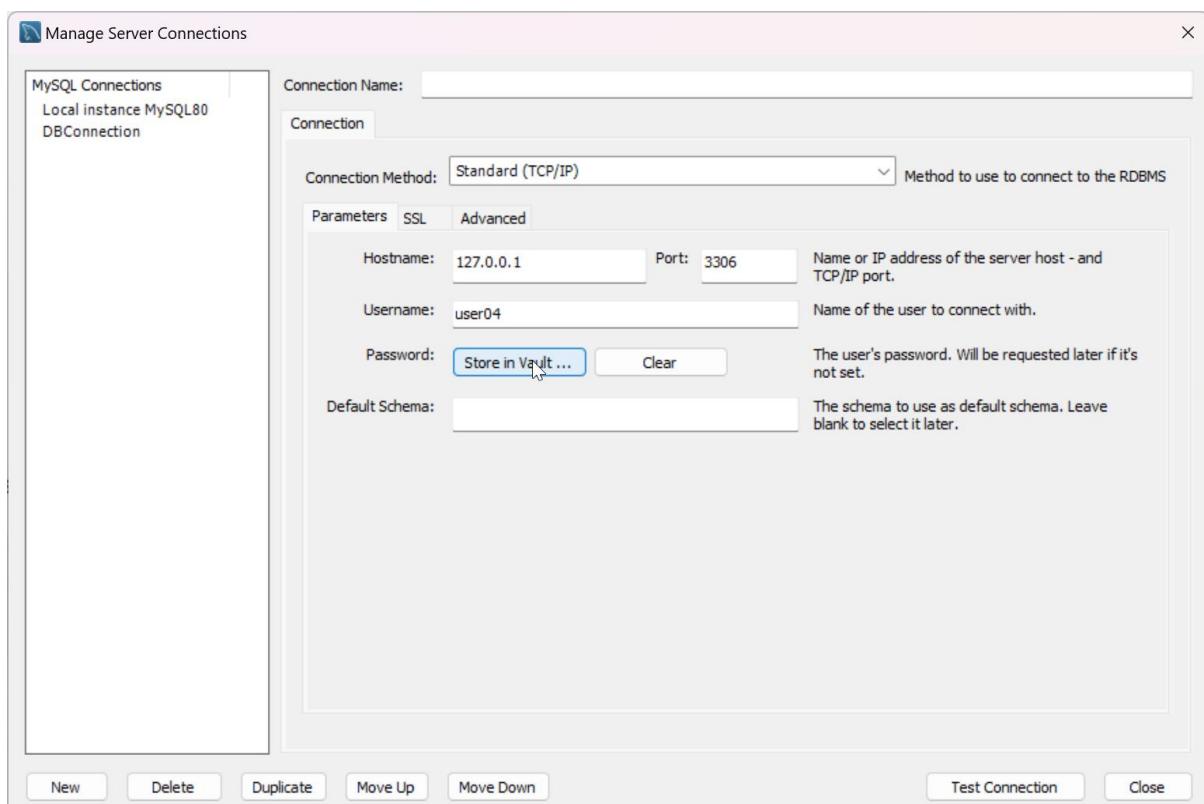
Output		
#	Time	Action
20	02:45:41	CREATE USER 'user04'@'localhost' IDENTIFIED BY 'U04@ser'
21	02:45:50	GRANT SELECT ON abc_automart.sales TO 'user04'@'localhost'

```

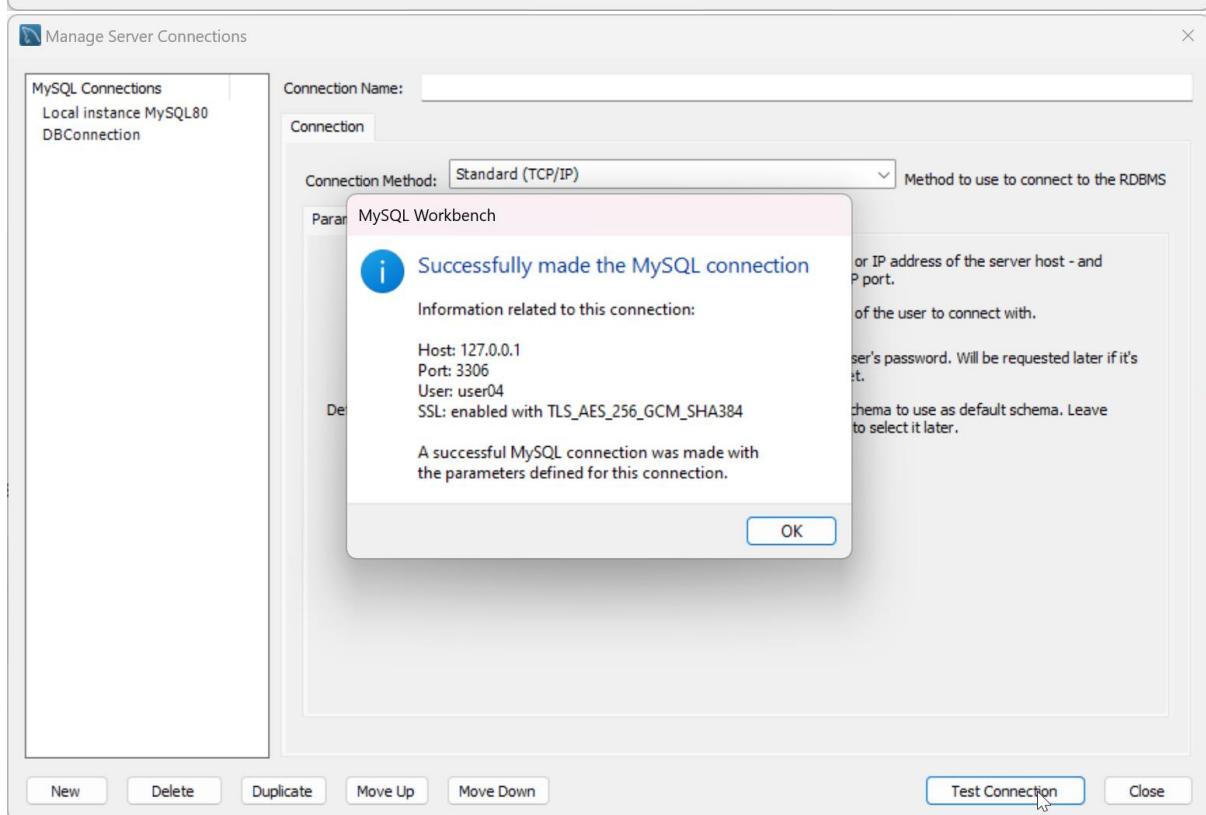
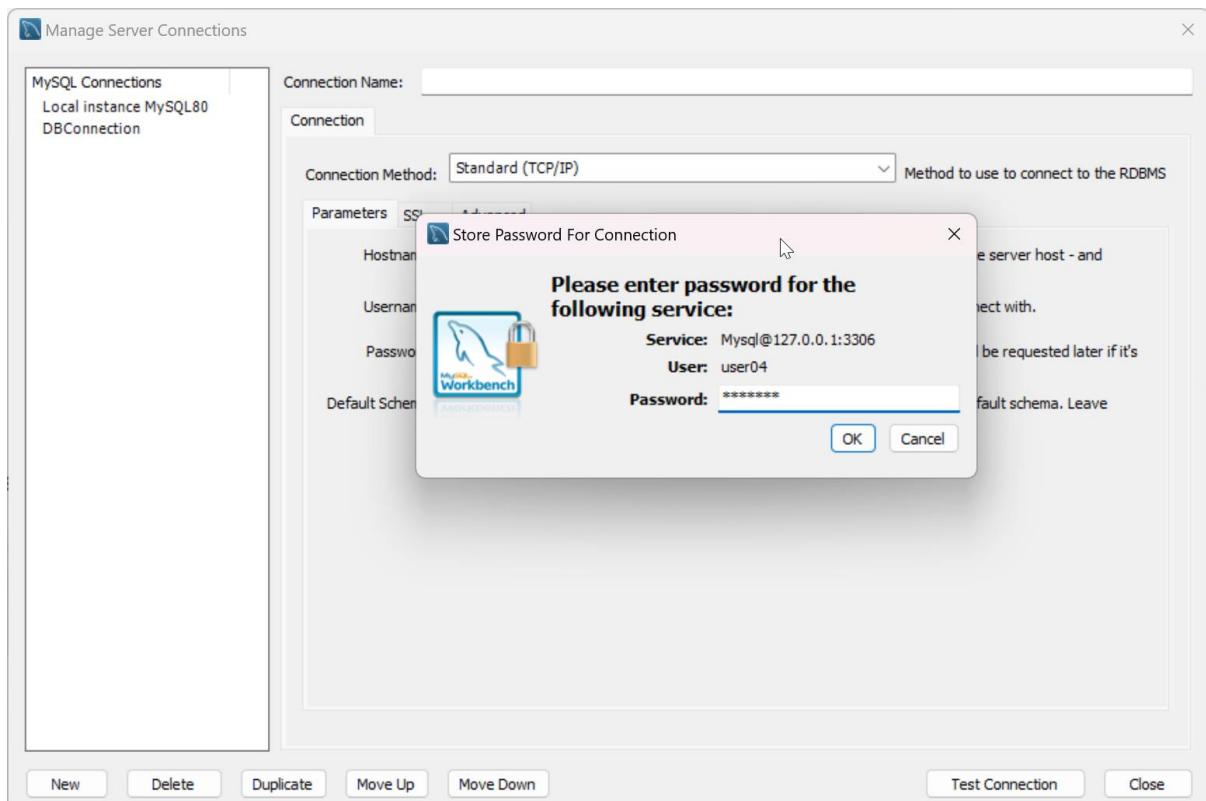
139  --
140  -- Using user id user04, retrieve sales information. Try to update selling price for sales on
141  -- 900006 to RN55K. What is the output when you try to do this transaction? Justify your
142  -- answer.
143

```

In order to do that we need to logout from current user.



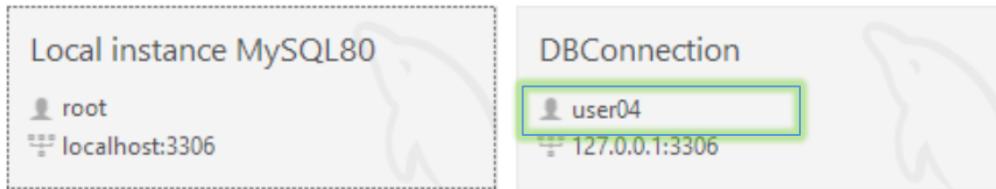
Enter user name "user04" and password "U04@ser"



**Test the connection and ready to go**

[Browse Documentation >](#)

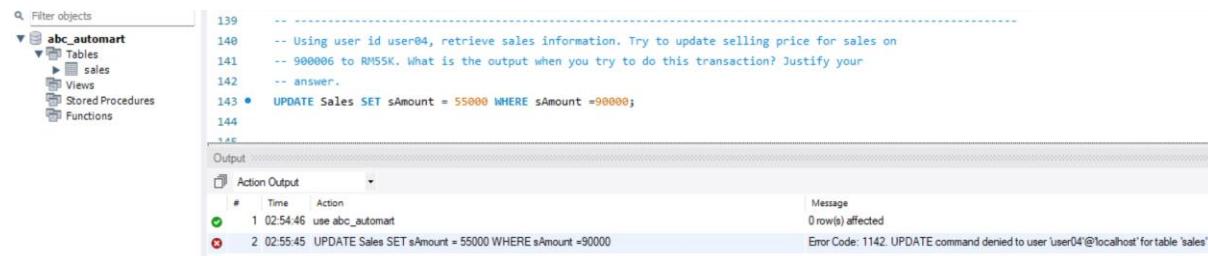
## MySQL Connections



The screenshot shows the MySQL Connections panel in MySQL Workbench. It displays two connections: "Local instance MySQL80" and "DBConnection". The "Local instance MySQL80" connection shows the "root" user and the "localhost:3306" port. The "DBConnection" connection shows the "user04" user and the "127.0.0.1:3306" port. The "user04" entry is highlighted with a green border.

We can see now its user04

Lets try to do the requirement:



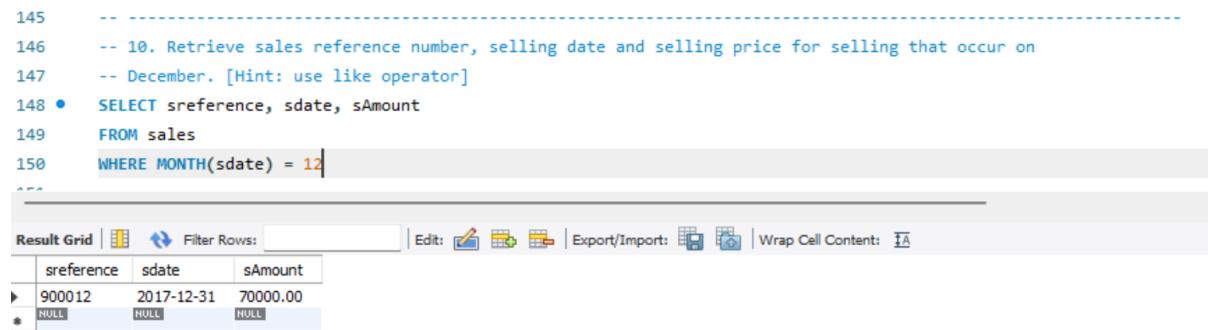
The screenshot shows the SQL editor in MySQL Workbench. The code being run is:

```
139 -- 
140 -- Using user id user04, retrieve sales information. Try to update selling price for sales on
141 -- 900006 to RM55K. What is the output when you try to do this transaction? Justify your
142 -- answer.
143 • UPDATE Sales SET sAmount = 55000 WHERE sAmount =90000;
144
```

The output shows the following actions:

#	Time	Action	Message
1	02:54:46	use abc_automart	0 row(s) affected
2	02:55:45	UPDATE Sales SET sAmount = 55000 WHERE sAmount =90000	Error Code: 1142. UPDATE command denied to user 'user04'@'localhost' for table 'sales'

Access privileges denied because user04 are allowed only to view the information in Sales table.

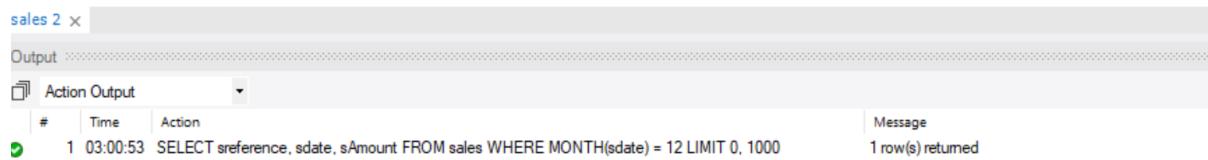


The screenshot shows the SQL editor in MySQL Workbench. The code being run is:

```
145 -- -----
146 -- 10. Retrieve sales reference number, selling date and selling price for selling that occur on
147 -- December. [Hint: use like operator]
148 • SELECT sreference, sdate, sAmount
149 FROM sales
150 WHERE MONTH(sdate) = 12
```

The output shows the results of the query in a result grid:

sreference	sdate	sAmount
900012	2017-12-31	70000.00
NULL	NULL	NULL



The screenshot shows the output of a query named "sales 2" in MySQL Workbench. The output shows the results of the query in a result grid:

sreference	sdate	sAmount
900012	2017-12-31	70000.00
NULL	NULL	NULL

The message area indicates "1 row(s) returned".

As we have only one month (December) so we got only one record. THANK YOU.