SQL Assignment-1

1. **Create Database e_commerce**
2. **Create following Tables:**

Customers:

SQL File 5*

```sql
create database e_commerce;

use e_commerce;

create table customers(customer_id  int auto_increment primary key,
   name varchar(50),
   email varchar(50),
   mobile varchar(15)
);
select * from customers;
```

Result Grid

customer_id	name	email	mobile
NULL	NULL	NULL	NULL

customers 1

Output

Action Output

#	Time	Action	Message
1	23:52:15	create database e_commerce	1 row(s) affected
2	23:55:30	use e_commerce	0 row(s) affected
3	23:55:35	create table customers(customer_id - int auto-increment primary key, name - varchar(50), email - varchar(50), mobile - varchar(15))	Error Code: 1064.
4	23:57:28	create table customers(customer_id int auto_increment primary key, name varchar(50), email varchar(50), mobile varchar(15))	0 row(s) affected
5	23:59:28	select * from customers LIMIT 0, 1000	0 row(s) returned

Products:

```
12 • create table products(id int,
13     name varchar(50) not null,
14     description varchar(200),
15     price  decimal(10, 2) not null,
16     category  varchar(50)
17   );
18 • select * from products;
```

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

id	name	description	price	category

products 2

Output

Action Output

#	Time	Action
1	23:52:15	create database e_commerce
2	23:55:30	use e_commerce
3	23:55:35	create table customers(customer_id - int auto-increment primary key, name - varchar(50), email - varchar(50), mobile - varchar(15)
4	23:57:28	create table customers(customer_id int auto_increment primary key, name varchar(50), email varchar(50), mobile varchar(15))
5	23:59:28	select * from customers LIMIT 0, 1000
6	00:04:59	create table products(id int, name varchar(50) not null, description varchar(200), price decimal(10, 2) not null, category varch.
7	00:05:18	select * from products LIMIT 0, 1000

3. Modify Tables(using Alter keyword):

a.Add not null on name and email in the Customers table

```
20 • ALTER TABLE customers
21   MODIFY COLUMN name VARCHAR(50) NOT NULL,
22   MODIFY COLUMN email VARCHAR(50) NOT NULL;
```

Output

Action Output

#	Time	Action
1	16:47:11	ALTER TABLE customers MODIFY COLUMN name VARCHAR(50) NOT NULL, MODIFY COLUMN email VARCHAR(50)

```sql
-- b. Add UNIQUE constraint on email in Customers table
ALTER TABLE Customers
ADD CONSTRAINT unique_email UNIQUE (email);

-- c. Add column age in Customers table
ALTER TABLE Customers
ADD COLUMN age INT;

-- d. Change column name from id to product_id in Products table
ALTER TABLE Products
CHANGE COLUMN id product_id INT;

-- e. Add primary key and auto increment on product_id in Products table
ALTER TABLE Products
MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY;

-- f. Change datatype of description from VARCHAR to TEXT in Products table
ALTER TABLE Products
MODIFY COLUMN description TEXT;
```

Output

Action Output

#	Time	Action
1	16:47:11	ALTER TABLE customers MODIFY COLUMN name VARCHAR(50) NOT NULL, MODIFY COLUMN email VARCHAR(50) NO
2	16:49:48	ALTER TABLE Customers ADD CONSTRAINT unique_email UNIQUE (email)
3	16:49:48	ALTER TABLE Customers ADD COLUMN age INT
4	16:49:48	ALTER TABLE Products CHANGE COLUMN id product_id INT
5	16:49:48	ALTER TABLE Products MODIFY COLUMN product_id INT AUTO_INCREMENT PRIMARY KEY
6	16:49:48	ALTER TABLE Products MODIFY COLUMN description TEXT

After modification tables:

```sql
select * from customers;
```

Result Grid | Filter Rows:

customer_id	name	email	mobile	age
NULL	NULL	NULL	NULL	NULL

```sql
select * from products;
```

Result Grid | Filter Rows:

product_id	name	description	price	category
NULL	NULL	NULL	NULL	NULL

4. Create table Order:

```sql
CREATE TABLE Orders (
    order_id INT AUTO_INCREMENT PRIMARY KEY,
    customer_id INT,
    product_id INT,
    quantity INT NOT NULL,
    order_date DATE NOT NULL,
    status ENUM('Pending', 'Success', 'Cancel'),
    payment_method ENUM('Credit', 'Debit', 'UPI'),
    total_amount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
```

Output

Action Output

#	Time	Action
1	17:15:12	CREATE TABLE Orders (order_id INT AUTO_INCREMENT PRIMARY KEY, cu

5. Modify Orders Table(using Alter keyword):

```sql
-- a. Rename table Order to Orders
ALTER TABLE Orders
RENAME TO Orders;

-- b. Set default value 'Pending' in status column
ALTER TABLE Orders
MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending';

-- c. Modify payment_method ENUM to add 'COD'
ALTER TABLE Orders
MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD');

-- d. Make product_id a foreign key
ALTER TABLE Orders
ADD CONSTRAINT fk_product FOREIGN KEY (product_id) REFERENCES Products(product_id);
```

Output

Action Output

#	Time	Action
1	17:17:47	ALTER TABLE Orders RENAME TO Orders
2	17:17:47	ALTER TABLE Orders MODIFY COLUMN status ENUM('Pending', 'Success', 'Cancel') DEFAULT 'Pending'
3	17:17:47	ALTER TABLE Orders MODIFY COLUMN payment_method ENUM('Credit', 'Debit', 'UPI', 'COD')
4	17:17:47	ALTER TABLE Orders ADD CONSTRAINT fk_product FOREIGN KEY (product_id) REFERENCES Products(product_id)

6. **Insert 20 sample records in all the tables.**

Customers:

```sql
INSERT INTO Customers (name, email, mobile, age) VALUES
('John ', 'john@example.com', '98765', 30),
('Alice ', 'alice@example.com', '99887', 28),
('Bob ', 'bob@example.com', '98712', 35),
('Emma ', 'emma@example.com', '91234', 26),
('Chris ', 'chris@example.com', '87654', 32),
('Davis', 'olivia@example.com', '90987', 27),
('Michael', 'michael@example.com', '92123', 40),
('Sophia ', 'sophia@example.com', '93456', 24),
('Daniel ', 'daniel@example.com', '90126', 29),
('Emily ', 'emily@example.com', '91239', 31),
('David ', 'david@example.com', '9456123789', 33),
('Ella ', 'ella@example.com', '9654321098', 23),
('Henry ', 'henry@example.com', '9784561230', 38),
('Isabella ', 'isabella@example.com', '9876541230', 25),
```

Output

Action Output

#	Time	Action
1	17:23:28	INSERT INTO Customers (name, email, mobile, age) VALUES ('John ', 'joh

customer_id	name	email	mobile	age
1	John	john@example.com	98765	30
2	Alice	alice@example.com	99887	28
3	Bob	bob@example.com	98712	35
4	Emma	emma@example.com	91234	26
5	Chris	chris@example.com	87654	32
6	Davis	olivia@example.com	90987	27
7	Michael	michael@example.com	92123	40
8	Sophia	sophia@example.com	93456	24
9	Daniel	daniel@example.com	90126	29
10	Emily	emily@example.com	91239	31
11	David	david@example.com	9456...	33
12	Ella	ella@example.com	9654...	23
13	Henry	henry@example.com	9784...	38
14	Isabella	isabella@example.com	9876...	25
15	James	james@example.com	9123...	37
16	Grace	grace@example.com	9998...	29
17	Lucas	lucas@example.com	9765...	34
18	Young	charlotte@example.com	9018...	27
19	King	mason@example.com	9554...	30
20	Green	amelia@example.com	9654...	22
NULL	NULL	NULL	NULL	NULL

Products :

esult Grid | Filter Rows: | Edit:

product_id	name	description	price	category
1	Lap...	High-end	75000.00	Electronics
2	Phone	Latest	45000.00	Electronics
3	Hea...	Noise-free	8000.00	Accessories
4	Watch	Fitness	12000.00	Wearable
5	Ca...	DSLR	30000.00	Photography
6	Tablet	Fast	50000.00	Electronics
7	Mon...	4K	21000.00	Electronics
8	Mouse	Wireless	2000.00	Accessories
9	Key...	Mechanical	5000.00	Accessories
10	Printer	Laser	25000.00	Office
11	Spe...	Bluetooth	8500.00	Audio
12	TV	Smart	60000.00	Electronics
13	Proj...	HD	6000.00	Office
14	Pow...	Fast	18000.00	Accessories
15	Rou...	Dual-band	8000.00	Networking
16	SSD	1TB	70000.00	Storage
17	Har...	2TB	3500.00	Storage
18	Tripod	Adjustable	10000.00	Photography
19	VR ...	Gaming	20000.00	Gaming
20	Drone	4K	12000.00	Photography
NULL	NULL	NULL	NULL	NULL

Orders:

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell

order_id	customer_id	product_id	quantity	order_date	status	payment_method	total_amount
21	1	3	2	2024-01-10	Success	Credit	16000.00
22	2	1	1	2024-01-15	Pending	Debit	75000.00
23	3	5	1	2024-02-05	Cancel	UPI	30000.00
24	4	2	1	2024-02-10	Success	COD	45000.00
25	5	7	3	2024-02-20	Success	Credit	21000.00
26	6	10	1	2024-03-01	Pending	Debit	25000.00
27	7	12	2	2024-03-15	Cancel	UPI	30000.00
28	8	8	1	2024-03-20	Success	COD	2000.00
29	9	6	1	2024-04-01	Pending	Credit	50000.00
30	10	4	1	2024-04-10	Success	Debit	12000.00
31	11	9	2	2024-04-15	Success	UPI	10000.00
32	12	14	1	2024-05-01	Cancel	COD	18000.00
33	13	16	1	2024-05-10	Pending	Credit	70000.00
34	14	11	1	2024-05-15	Success	Debit	85000.00
35	15	13	1	2024-06-01	Pending	UPI	6000.00
36	16	15	2	2024-06-10	Success	COD	8000.00
37	17	18	1	2024-06-20	Cancel	Credit	10000.00
38	18	19	1	2024-07-01	Success	Debit	20000.00
39	19	17	1	2024-07-05	Pending	UPI	3500.00
40	20	20	1	2024-07-10	Success	COD	12000.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

7. **Perform following queries:**

a. Count the number of products as product_count in each category.

```
SELECT category, COUNT(*) AS product_count
FROM Products
GROUP BY category;
```

esult Grid | Filter Rows: | Export:

category	product_count
Electronics	5
Accessories	4
Wearable	1
Photography	3
Office	2
Audio	1
Networking	1
Storage	2
Gaming	1

b. Retrieve all products that belong to the 'Electronics' category, have a price between $50 and $500, and whose name contains the letter 'a'.

```
SELECT *
FROM Products
WHERE category = 'Electronics'
AND price BETWEEN 50 AND 500
AND name LIKE '%a%';
```

sult Grid | Filter Rows: | Edit:

product_id	name	description	price	category
NULL	NULL	NULL	NULL	NULL

c. Get the top 5 most expensive products in the 'Electronics' category, skipping the first 2.

```
SELECT *
FROM Products
WHERE category = 'Electronics'
ORDER BY price DESC
LIMIT 5 OFFSET 2;
```

Result Grid | Filter Rows: | Edit:

product_id	name	description	price	category
6	Tablet	Fast	50000.00	Electronics
2	Phone	Latest	45000.00	Electronics
7	Mon...	4K	21000.00	Electronics
NULL	NULL	NULL	NULL	NULL

d.Retrieve customers who have not placed any orders.

```
SELECT *
FROM Customers
WHERE customer_id NOT IN (SELECT DISTINCT customer_id FROM Orders);
```

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

customer_id	name	email	mobile	age
NULL	NULL	NULL	NULL	NULL

e. Find the average total amount spent by each customer.

```
SELECT customer_id, AVG(total_amount) AS avg_spent
FROM Orders
GROUP BY customer_id;
```

esult Grid | Filter Rows: | Export: | Wrap Cell C

customer_id	avg_spent
1	16000.000000
2	75000.000000
3	30000.000000
4	45000.000000
5	21000.000000
6	25000.000000
7	30000.000000
8	2000.000000
9	50000.000000
10	12000.000000
11	10000.000000
12	18000.000000
13	70000.000000
14	85000.000000
15	6000.000000
16	8000.000000
17	10000.000000
18	20000.000000
19	3500.000000
20	12000.000000

f. Get the products that have a price less than the average price of all products.

```
SELECT *
FROM Products
WHERE price < (SELECT AVG(price) FROM Products);
```

esult Grid | Filter Rows: | Edit:

product_id	name	description	price	category
3	Hea...	Noise-free	8000.00	Accessories
4	Watch	Fitness	12000.00	Wearable
7	Mon...	4K	21000.00	Electronics
8	Mouse	Wireless	2000.00	Accessories
9	Key...	Mechanical	5000.00	Accessories
11	Spe...	Bluetooth	8500.00	Audio
13	Proj...	HD	6000.00	Office
14	Pow...	Fast	18000.00	Accessories
15	Rou...	Dual-band	8000.00	Networking
17	Har...	2TB	3500.00	Storage
18	Tripod	Adjustable	10000.00	Photography
19	VR ...	Gaming	20000.00	Gaming
20	Drone	4K	12000.00	Photography

g. Calculate the total quantity of products ordered by each customer:

```
SELECT customer_id, SUM(quantity) AS total_quantity
FROM Orders
GROUP BY customer_id;
```

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

customer_id	total_quantity
1	2
2	1
3	1
4	1
5	3
6	1
7	2
8	1
9	1
10	1
11	2
12	1
13	1
14	1
15	1
16	2
17	1
18	1
19	1
20	1

h. List all orders along with customer name and product name.

```
SELECT O.order_id, C.name AS customer_name, P.name AS product_name, O.quantity, O.total_amount
FROM Orders O
JOIN Customers C ON O.customer_id = C.customer_id
JOIN Products P ON O.product_id = P.product_id;
```

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

order_id	customer_name	product_name	quantity	total_amount
21	John	Headphones	2	16000.00
22	Alice	Laptop	1	75000.00
23	Bob	Camera	1	30000.00
24	Emma	Phone	1	45000.00
25	Chris	Monitor	3	21000.00
26	Davis	Printer	1	25000.00
27	Michael	TV	2	30000.00
28	Sophia	Mouse	1	2000.00
29	Daniel	Tablet	1	50000.00
30	Emily	Watch	1	12000.00
31	David	Keyboard	2	10000.00
32	Ella	Powerbank	1	18000.00
33	Henry	SSD	1	70000.00
34	Isabella	Speaker	1	85000.00
35	James	Projector	1	6000.00
36	Grace	Router	2	8000.00
37	Lucas	Tripod	1	10000.00
38	Young	VR Headset	1	20000.00
39	King	Hard Disk	1	3500.00
40	Green	Drone	1	12000.00

i. Find products that have never been ordered.

```
SELECT *
FROM Products
WHERE product_id NOT IN (SELECT DISTINCT product_id FROM Orders);
```

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

product_id	name	description	price	category
NULL	NULL	NULL	NULL	NULL