

NACTAR MLDS Module 11 Exam Solution. Created by Md. Iquball Hossain (Roll-12, Batch-03)
Topics: SQL Query for MySQL Database

Problem 1: SELECT and WHERE

Ques: You have a table named `students` with columns `student_id`, `name`, `age`, and `gender`. Write a query to select the names of female students who are below 25 years of age.

SELECT name

FROM students WHERE gender = 'female' AND age < 25;

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'students' selected under the 'mlds_11' database. The main editor shows a SQL query with two statements: a SELECT statement to view all data in the 'students' table, and a WHERE clause query to filter for female students under 25. The 'Result Grid' shows the output of the first query, listing 8 students with their IDs, names, ages, and genders. The 'Output' pane at the bottom shows the execution log, indicating that the first query returned 8 rows and the second query returned 3 rows.

student_id	name	age	gender
1	rafic	25	male
2	tania	15	female
3	tonni	17	female
4	iquball	23	male
5	li	28	female
6	ovi	25	male
7	ripon	30	male
8	mahi	20	female

This screenshot shows the same MySQL Workbench interface, but with the 'Automatic context help is disabled' message visible on the right. The 'Result Grid' now only displays the names of the three female students: 'tania', 'tonni', and 'mahi'. The 'Output' pane shows the execution log, including the duration and fetch time for each query. The 'Context Help' and 'Snippets' buttons are visible at the bottom right.

name
tania
tonni
mahi

NACTAR MLDS Module 11 Exam Solution. Created by Md. Iquball Hossain (Roll-12, Batch-03)
Topics: SQL Query for MySQL Database

Problem 2: ORDER BY, GROUP BY, and AGGREGATE FUNCTIONS

Ques: Consider a table named `sales` with columns `product_id`, `product_name`, `category`, and `sales_amount`. Write a query to find the total sales amount for each category, and display the results in descending order of total sales amount.

```
SELECT category, SUM (sales_amount) AS total_sales
FROM sales
GROUP BY category
ORDER BY total_sales DESC;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected. The main editor contains the following SQL query:

```
1 • SELECT * FROM mlds_11.sales;
2 • SELECT category, SUM(sales_amount) AS total_sales
3 FROM sales
4 GROUP BY category
5 ORDER BY total_sales DESC
6 LIMIT 0, 10;
7
```

Below the query editor, the 'Result Grid' shows the output of the query:

product_id	product_name	category	sales_amount
1	rice	food	700.00
2	mobile	device	15000.00
3	shirt	dress	2000.00

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected. The main editor contains the following SQL query:

```
1 • SELECT * FROM mlds_11.sales;
2 • SELECT category, SUM(sales_amount) AS total_sales
3 FROM sales
4 GROUP BY category
5 ORDER BY total_sales DESC
6 LIMIT 0, 10;
7
```

Below the query editor, the 'Result Grid' shows the output of the query:

category	total_sales
device	15000.00
dress	2000.00
food	700.00

NACTAR MLDS Module 11 Exam Solution. Created by Md. Iquball Hossain (Roll-12, Batch-03)
Topics: SQL Query for MySQL Database

Problem 3: JOIN, WHERE, and LIKE

Ques: Suppose you have two tables named `employees` and `departments`, where `employees` contains columns `employee_id`, `name`, `department_id`, and `salary`, and `departments` contains columns `department_id` and `department_name`. Write a query to select the names and salaries of employees who work in the 'Sales' department and whose salaries are greater than \$50000.

```
SELECT e.name, e.salary
FROM employees e
JOIN departments d ON e.department_id = d.department_id
WHERE d.department_name LIKE '%Sales%' AND e.salary > 50000;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected, showing tables 'departments', 'employees', 'sales', and 'students'. The main editor window contains the following SQL query:

```
1 • SELECT * FROM mlds_11.employees;
2 • SELECT e.name, e.salary
3 FROM employees e
4 JOIN departments d ON e.department_id = d.department_id
5 WHERE d.department_name LIKE '%Sales%' AND e.salary > 50000;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query:

employee_id	department_id	name	salary
1	2	rafiq	50000.00
2	3	karim	25000.00
3	1	sumon	75000.00
4	4	jabbar	55000.00
5	2	sifat	60000.00

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected, showing tables 'departments', 'employees', 'sales', and 'students'. The main editor window contains the following SQL query:

```
1 SELECT * FROM mlds_11.employees;
2 SELECT e.name, e.salary
3 FROM employees e
4 JOIN departments d ON e.department_id = d.department_id
5 WHERE d.department_name LIKE '%Sales%' AND e.salary > 50000;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query:

name	salary
sifat	60000.00

NACTAR MLDS Module 11 Exam Solution. Created by Md. Iquball Hossain (Roll-12, Batch-03)
Topics: SQL Query for MySQL Database

Problem 4: NOT, Wildcards, and LIKE

Ques: Assume you have a table named `products` with columns `product_id`, `product_name`, and `price`. Write a query to select the names of products that do not contain the word 'cheap' in their names.

```
SELECT product_name  
FROM products  
WHERE product_name NOT LIKE '%cheap%';
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected. The main editor shows a SQL query: `SELECT * FROM mlds_11.products;`, `SELECT product_name`, `FROM products`, and `WHERE product_name NOT LIKE '%cheap%';`. The 'Result Grid' at the bottom displays the following data:

product_id	product_name	price
1	cheaper	1000.00
2	sugar	200.00
3	salt	500.00
4	chili	1000.00

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected. The main editor shows a SQL query: `SELECT * FROM mlds_11.products;`, `SELECT product_name`, `FROM products`, and `WHERE product_name NOT LIKE '%cheap%';`. The 'Result Grid' at the bottom displays the following data:

product_name
sugar
salt
chili

NACTAR MLDS Module 11 Exam Solution. Created by Md. Iquball Hossain (Roll-12, Batch-03)
Topics: SQL Query for MySQL Database

Problem 5: Views and Joins

Ques: Consider two tables: `orders` with columns `order_id`, `customer_id`, and `order_date`, and `customers` with columns `customer_id`, `customer_name`, and `city`. Create a view named `customer_orders` that displays the customer name, order ID, and order date for each order, along with the city of the customer.

```
CREATE VIEW customer_orders AS  
SELECT c.customer_name, o.order_id, o.order_date, c.city  
FROM orders o  
JOIN customers c ON o.customer_id = c.customer_id;
```

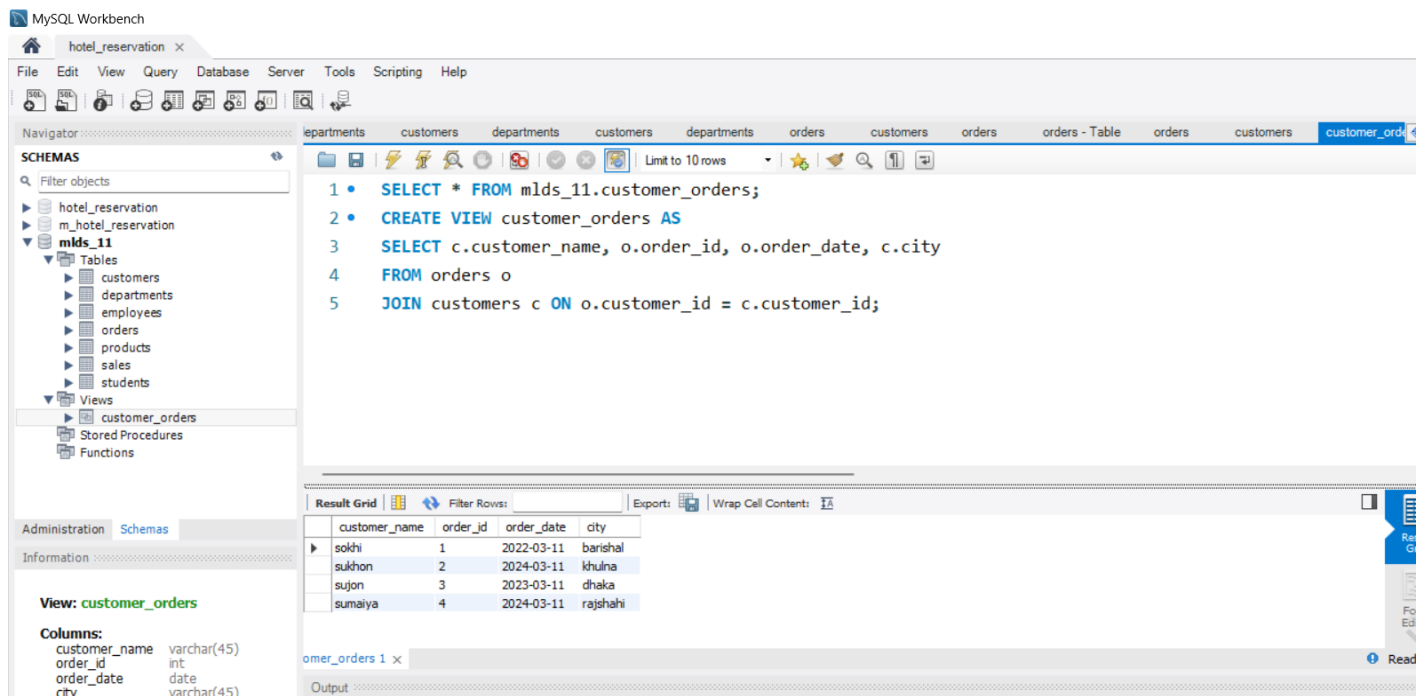
The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the database structure, including the 'mlds_11' database with tables 'customers', 'departments', 'employees', 'orders', 'products', 'sales', and 'students'. The 'orders' table is selected, and its structure is shown in the 'Information' pane: **Table: orders**, columns: **order_id** (int AI PK), **customer_id** (int), **order_date** (date). The main query editor shows the query: `SELECT * FROM mlds_11.orders;`. The 'Result Grid' displays the following data:

order_id	customer_id	order_date
1	2	2022-03-11
2	3	2024-03-11
3	1	2023-03-11
4	4	2024-03-11

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays the database structure, including the 'mlds_11' database with tables 'customers', 'departments', 'employees', 'orders', 'products', 'sales', and 'students'. The 'customers' table is selected, and its structure is shown in the 'Information' pane: **Table: customers**, columns: **customer_id** (int AI PK), **customer_name** (varchar(45)), **city** (varchar(45)). The main query editor shows the query: `SELECT * FROM mlds_11.customers;`. The 'Result Grid' displays the following data:

customer_id	customer_name	city
1	sujon	dhaka
2	sokhi	barishal
3	sukhon	khulna
4	sumaiya	rajshahi

NACTAR MLDS Module 11 Exam Solution. Created by Md. Iquball Hossain (Roll-12, Batch-03)
Topics: SQL Query for MySQL Database



MySQL Workbench interface showing the creation of a view named `customer_orders`. The SQL editor contains the following queries:

```
1 • SELECT * FROM mlds_11.customer_orders;
2 • CREATE VIEW customer_orders AS
3   SELECT c.customer_name, o.order_id, o.order_date, c.city
4   FROM orders o
5   JOIN customers c ON o.customer_id = c.customer_id;
```

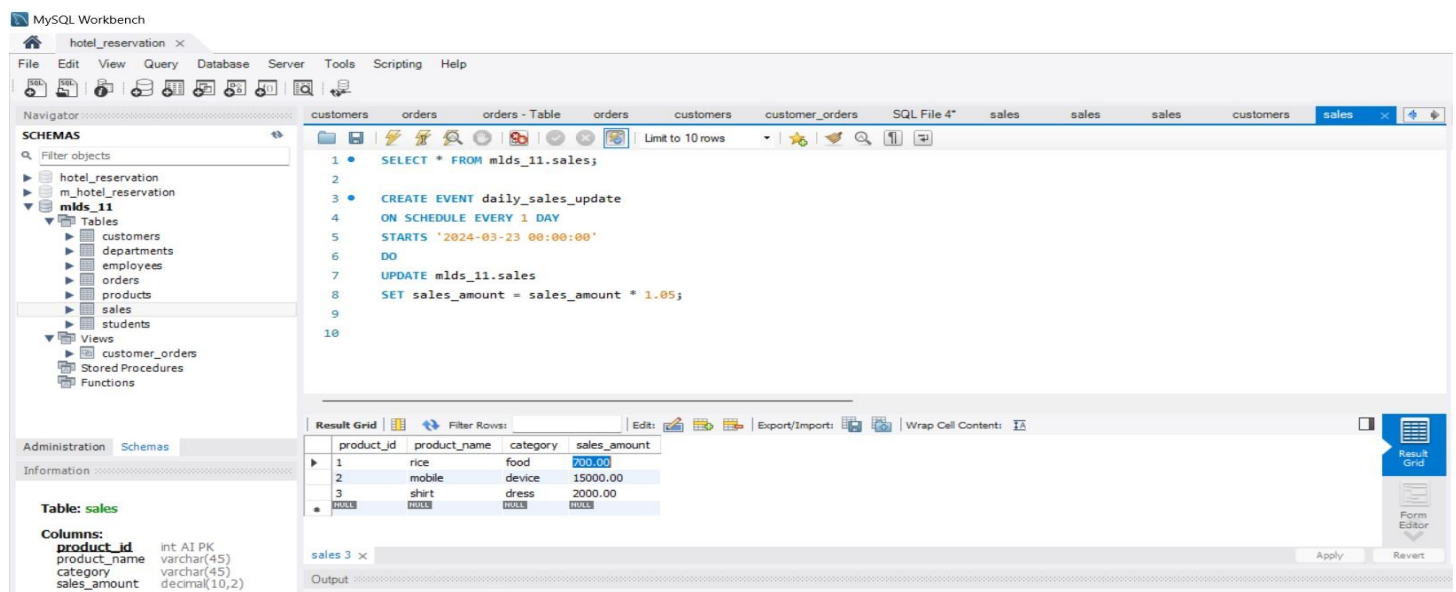
The Result Grid displays the data for the `customer_orders` view:

customer_name	order_id	order_date	city
sokhi	1	2022-03-11	barishal
sukhon	2	2024-03-11	khulna
sujon	3	2023-03-11	dhaka
sumaiya	4	2024-03-11	rajshahi

Problem 6: Event

Ques: Design an event in MySQL that runs every day at midnight (12 AM) and updates the `sales` table (Problem 2) by increasing the sales amount of each product by 5%.

```
CREATE EVENT daily_sales_update
ON SCHEDULE EVERY 1 DAY
STARTS 'YYYY-MM-DD 00:00:00'
DO
UPDATE sales
SET sales_amount = sales_amount * 1.05;
```



MySQL Workbench interface showing the creation of an event named `daily_sales_update`. The SQL editor contains the following queries:

```
1 • SELECT * FROM mlds_11.sales;
2 •
3 • CREATE EVENT daily_sales_update
4   ON SCHEDULE EVERY 1 DAY
5   STARTS '2024-03-23 00:00:00'
6   DO
7   UPDATE mlds_11.sales
8   SET sales_amount = sales_amount * 1.05;
9 •
10 •
```

The Result Grid displays the data for the `sales` table:

product_id	product_name	category	sales_amount
1	rice	food	700.00
2	mobile	device	15000.00
3	shirt	dress	2000.00
4	NULL	NULL	NULL

Problem 7: Normalization

Ques: You have a table named `customers` with columns `customer_id`, `customer_name`, `address`, `city`, and `country`. Identify any normalization issues in this table and propose a normalized schema to address them.

```
CREATE TABLE customers (  
  customer_id INT PRIMARY KEY,  
  customer_name VARCHAR(255)  
);  
  
CREATE TABLE addresses (  
  address_id INT PRIMARY KEY,  
  customer_id INT,  
  address VARCHAR(255),  
  city VARCHAR(100),  
  country VARCHAR(100),  
  FOREIGN KEY (customer_id) REFERENCES customers(customer_id)  
);
```

MySQL Workbench

hotel_reservation x

File Edit View Query Database Server Tools Scripting Help

Navigator: omers - Table addresses customers customers - Table addresses addresses - Table addresses addresses - Table addresses addresses - Table addresses

SCHEMAS

Filter objects

- hotel_reservation
- m_hotel_reservation
- mlds_11**
 - Tables
 - addresses
 - customers
 - departments
 - employees
 - orders
 - products
 - sales
 - students
 - Views
 - customer_orders
 - Stored Procedures
 - Functions

Administration Schemas

Information

Table: addresses

Columns:
address_id int PK
customer_id int
address varchar(255)
city varchar(100)
country varchar(100)

Table Name: addresses Schema: mlds_11

Charset/Collation: utf8mb4 utf8mb4_0900_ai_ci Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
address_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
customer_id	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
address	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
city	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
country	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

Column Name: Data Type: Charset/Collation: Default Charset Default Collation Default: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Comments:

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

Output: Action Output

Problem 8: GROUP BY and HAVING

Ques: Consider a table named `orders` with columns `order_id`, `customer_id`, `order_date`, and `total_amount`. Write a query to find the total number of orders placed by each customer who has placed more than 5 orders.

```
SELECT customer_id, COUNT(order_id) AS total_orders
FROM orders
GROUP BY customer_id
HAVING COUNT(order_id) > 5;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'mlds_11' selected, showing tables like 'orders'. The main editor contains the following SQL query:

```
1 • SELECT * FROM mlds_11.orders;
2 • SELECT customer_id, COUNT(order_id) AS total_orders
3   FROM orders
4  GROUP BY customer_id
5  HAVING COUNT(order_id) > 5;
```

Below the query editor, the 'Result Grid' is visible, showing the results of the query:

customer_id	total_orders
3	6

The bottom left pane shows the 'Table: orders' structure:

Columns:

- order_id: int AI PK
- customer_id: int
- order_date: date
- total_amount: decimal(10,2)

The bottom right pane shows the 'Output' tab with 'Result 4' selected, indicating the query was executed successfully.