

NAME:ASHWINI BASAVARAJ SIMPAGER

1) CREATING TABLE

CREATE TABLE employees

(emp_id INT PRIMARY

KEY,

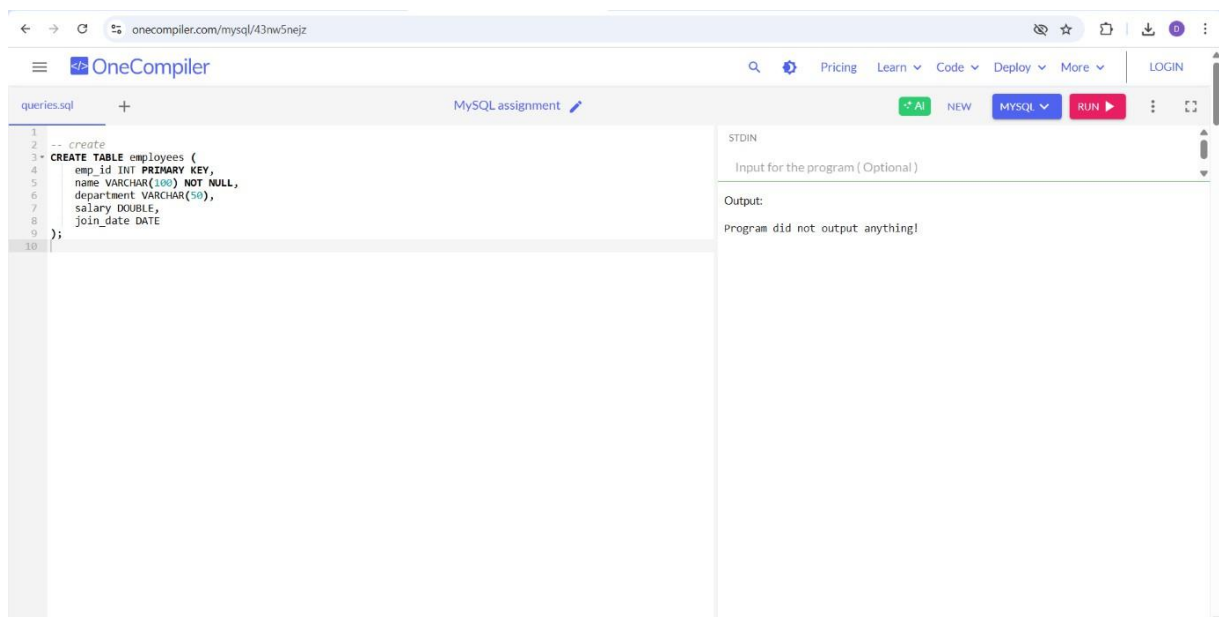
name VARCHAR(100) NOT NULL,

department VARCHAR(50),

salary DOUBLE,

join_date DATE

);



2) INSERT QUERY

INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES

(201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),

(202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),

(203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),

(204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),

(205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');

The screenshot shows the OneCompiler interface for a MySQL assignment. The code editor contains the following SQL queries:

```
1 -- create
2
3 CREATE TABLE employees (
4   emp_id INT PRIMARY KEY,
5   name VARCHAR(100) NOT NULL,
6   department VARCHAR(50),
7   salary DOUBLE,
8   join_date DATE
9 );
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
```

The right-hand side of the interface shows the execution results. The 'STDIN' section is empty. The 'Output' section displays the message: "Program did not output anything!".

3) SELECT QUERY

- SELECT * FROM employees;

The screenshot shows the OneCompiler interface for a MySQL assignment. The code editor contains the following SQL queries:

```
1 -- create
2
3 CREATE TABLE employees (
4   emp_id INT PRIMARY KEY,
5   name VARCHAR(100) NOT NULL,
6   department VARCHAR(50),
7   salary DOUBLE,
8   join_date DATE
9 );
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
```

The right-hand side of the interface shows the execution results. The 'STDIN' section is empty. The 'Output' section displays the following table:

emp_id	name	department	salary	join_date
201	Rahul Verma	HR	47000	2022-04-12
202	Sneha Kapoor	IT	82000	2021-02-18
203	Amit Sharma	Finance	61000	2020-09-07
204	Neha Joshi	IT	87000	2023-01-20
205	Karan Mehta	Sales	53000	2022-07-30

- SELECT name, department FROM employees;

The screenshot shows the OneCompiler interface with a MySQL assignment. The code in the editor is as follows:

```

1  -- create
2  CREATE TABLE employees (
3    emp_id INT PRIMARY KEY,
4    name VARCHAR(100) NOT NULL,
5    department VARCHAR(50),
6    salary DOUBLE,
7    join_date DATE
8  );
9
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
17
18 SELECT name, department FROM employees;
19
20

```

The output on the right shows the result of the query:

name	department
Rahul Verma	HR
Sneha Kapoor	IT
Amit Sharma	Finance
Neha Joshi	IT
Karan Mehta	Sales

- SELECT * FROM employees WHERE department = 'IT';

The screenshot shows the OneCompiler interface with a MySQL assignment. The code in the editor is as follows:

```

1  -- create
2  CREATE TABLE employees (
3    emp_id INT PRIMARY KEY,
4    name VARCHAR(100) NOT NULL,
5    department VARCHAR(50),
6    salary DOUBLE,
7    join_date DATE
8  );
9
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
17
18 SELECT * FROM employees WHERE department = 'IT';
19
20

```

The output on the right shows the result of the query:

emp_id	name	department	salary	join_date
202	Sneha Kapoor	IT	82000	2021-02-18
204	Neha Joshi	IT	87000	2023-01-20

4) AND, IN BETWEEN & LIKE

- SELECT * FROM employees WHERE department = 'IT' AND salary > 75000;

Pricing
Learn
Code
Deploy
More

LOGIN

queries.sql

+

MySQL assignment

NEW

MYSQL

RUN

⋮

🔍

```

1
2 -- create
3 CREATE TABLE employees (
4   emp_id INT PRIMARY KEY,
5   name VARCHAR(100) NOT NULL,
6   department VARCHAR(50),
7   salary DOUBLE,
8   join_date DATE
9 );
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
17
18 SELECT * FROM employees WHERE department = 'IT' AND salary > 75000;
19
20
21

```

STDIN

Input for the program (Optional)

Output:

emp_id	name	department	salary	join_date
202	Sneha Kapoor	IT	82000	2021-02-18
204	Neha Joshi	IT	87000	2023-01-20

- SELECT * FROM employees WHERE department IN ('IT', 'Finance');

Pricing
Learn
Code
Deploy
More

LOGIN

queries.sql

+

MySQL assignment

NEW

MYSQL

RUN

⋮

🔍

```

1
2 -- create
3 CREATE TABLE employees (
4   emp_id INT PRIMARY KEY,
5   name VARCHAR(100) NOT NULL,
6   department VARCHAR(50),
7   salary DOUBLE,
8   join_date DATE
9 );
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
17
18 SELECT * FROM employees WHERE department IN ('IT', 'Finance');
19
20
21
22

```

STDIN

Input for the program (Optional)

Output:

emp_id	name	department	salary	join_date
202	Sneha Kapoor	IT	82000	2021-02-18
203	Amit Sharma	Finance	61000	2020-09-07
204	Neha Joshi	IT	87000	2023-01-20

- SELECT * FROM employees WHERE salary BETWEEN 50000 AND 70000;

The screenshot shows a MySQL assignment in OneCompiler. The code defines an 'employees' table with columns: emp_id (INT PRIMARY KEY), name (VARCHAR(100) NOT NULL), department (VARCHAR(50)), salary (DOUBLE), and join_date (DATE). It then inserts five records into the table. Finally, a SELECT query is executed to display all records, ordered by salary in descending order.

```

1  -- create
2  CREATE TABLE employees (
3    emp_id INT PRIMARY KEY,
4    name VARCHAR(100) NOT NULL,
5    department VARCHAR(50),
6    salary DOUBLE,
7    join_date DATE
8  );
9
10
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
17
18 SELECT * FROM employees
19 ORDER BY salary DESC;
20
21
22

```

Output:

emp_id	name	department	salary	join_date
204	Neha Joshi	IT	87000	2023-01-20
202	Sneha Kapoor	IT	82000	2021-02-18
203	Amit Sharma	Finance	61000	2020-09-07
205	Karan Mehta	Sales	53000	2022-07-30
201	Rahul Verma	HR	47000	2022-04-12

6) UPDATE QUERY

- SELECT department, AVG(salary) AS avg_salary
FROM employees
GROUP BY department;

The screenshot shows a MySQL assignment in OneCompiler. The code defines an 'employees' table with columns: emp_id (INT PRIMARY KEY), name (VARCHAR(100) NOT NULL), department (VARCHAR(50)), salary (DOUBLE), and join_date (DATE). It then inserts five records into the table. Finally, a SELECT query is executed to display the average salary for each department.

```

1  -- 1. Create the employees table
2  CREATE TABLE employees (
3    emp_id INT PRIMARY KEY,
4    name VARCHAR(100) NOT NULL,
5    department VARCHAR(50),
6    salary DOUBLE,
7    join_date DATE
8  );
9
10
11 -- 2. Insert initial employee data
12 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
13 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
14 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
15 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
16 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
17 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
18
19 SELECT department, AVG(salary) AS avg_salary
20 FROM employees
21 GROUP BY department;
22
23
24

```

Output:

department	avg_salary
HR	47000
IT	84500
Finance	61000
Sales	53000

- SELECT department, COUNT(*) AS emp_count
FROM employees
GROUP BY department
HAVING COUNT(*) > 1;

OneCompiler

pricingLearnCodeDeployMore

LOG IN

queries.sql43nw9us9b

NEWMySQLRUN

```
1 -- 1. Create the employees table
2 CREATE TABLE employees (
3     emp_id INT PRIMARY KEY,
4     name VARCHAR(100) NOT NULL,
5     department VARCHAR(50),
6     salary DOUBLE,
7     join_date DATE
8 );
9
10 -- 2. Insert initial employee data
11 INSERT INTO employees (emp_id, name, department, salary, join_date) VALUES
12 (201, 'Rahul Verma', 'HR', 47000, '2022-04-12'),
13 (202, 'Sneha Kapoor', 'IT', 82000, '2021-02-18'),
14 (203, 'Amit Sharma', 'Finance', 61000, '2020-09-07'),
15 (204, 'Neha Joshi', 'IT', 87000, '2023-01-20'),
16 (205, 'Karan Mehta', 'Sales', 53000, '2022-07-30');
17
18 SELECT department, COUNT(*) AS emp_count
19 FROM employees
20 GROUP BY department
21 HAVING COUNT(*) > 1;
```

STDIN

Input for the program (Optional)

Output:

department	emp_count
IT	2