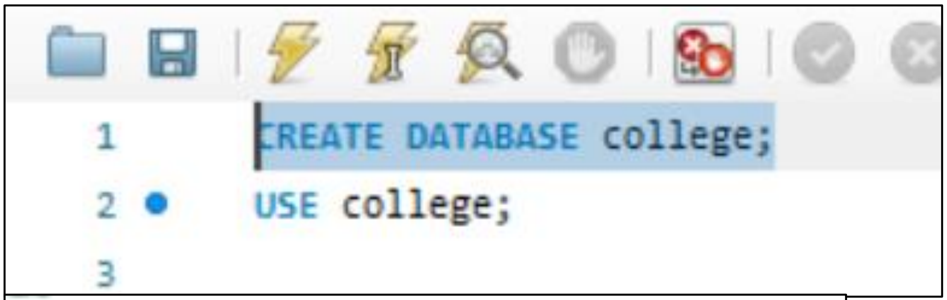
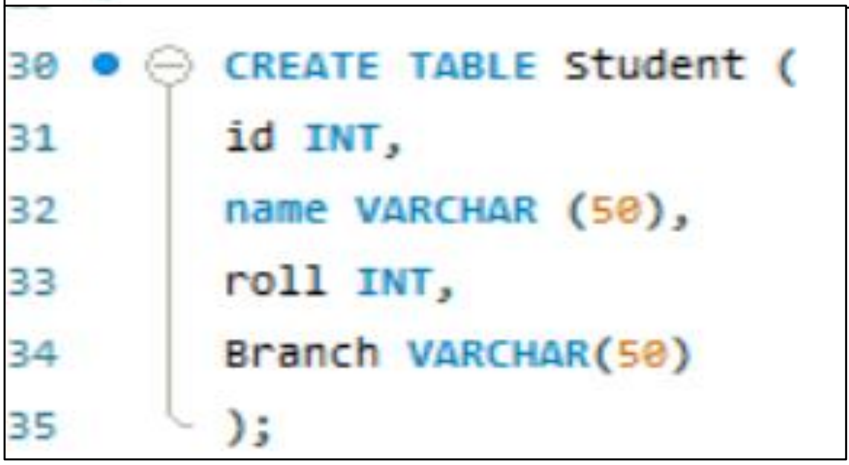
<2> To perform following SQL activity: a) Creating a database b) Creating Tables (With and Without Constraints) c) Inserting Record in table

a) Create Database



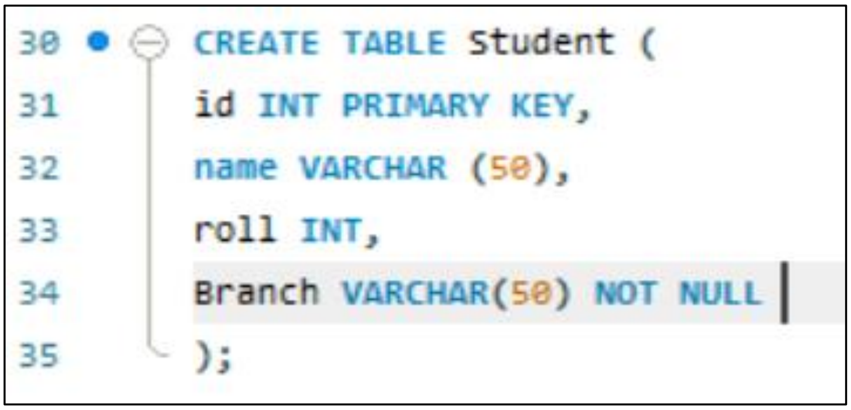
```sql
CREATE DATABASE college;
USE college;
```

b) Create Tables Without constraints



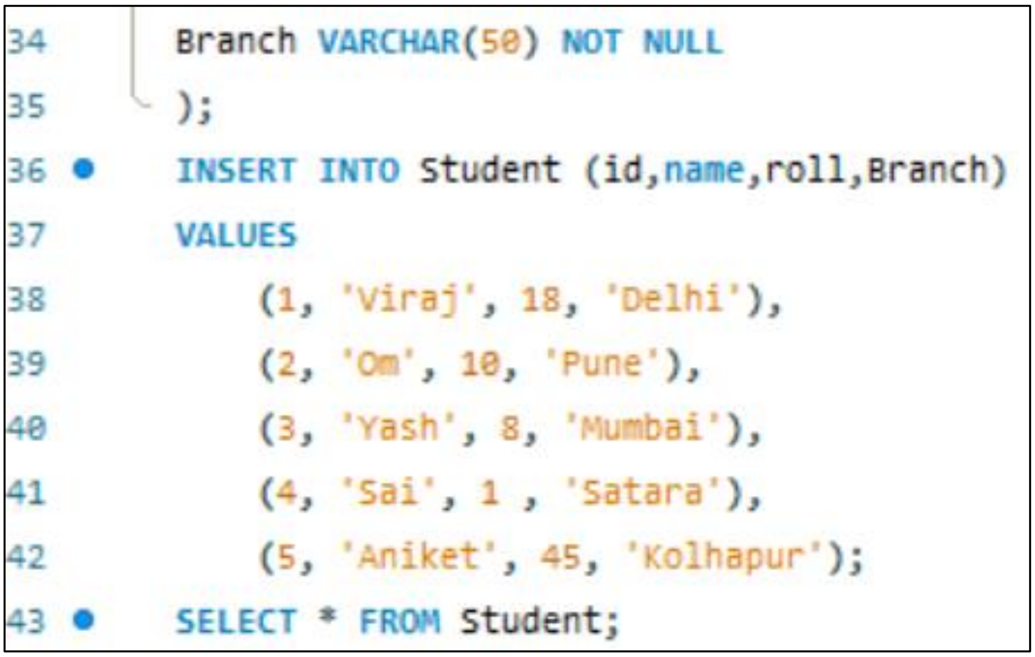
```sql
CREATE TABLE Student (
id INT,
name VARCHAR (50),
roll INT,
Branch VARCHAR(50)
);
```

b) Create Tables With constraints



```sql
CREATE TABLE Student (
id INT PRIMARY KEY,
name VARCHAR (50),
roll INT,
Branch VARCHAR(50) NOT NULL
);
```

c) Insert Record



```sql
Branch VARCHAR(50) NOT NULL
);
INSERT INTO Student (id,name,roll,Branch)
VALUES
     (1, 'Viraj', 18, 'Delhi'),
     (2, 'Om', 10, 'Pune'),
     (3, 'Yash', 8, 'Mumbai'),
     (4, 'Sai', 1 , 'Satara'),
     (5, 'Aniket', 45, 'Kolhapur');
SELECT * FROM Student;
```

3. To Perform the following: a. Viewing all databases, Viewing all Tables in a Database, Updating/Deleting Records in a Table

```
CREATE DATABASE Company;

USE Company;

CREATE TABLE employees (
   employee_id INT PRIMARY KEY,
   first_name VARCHAR(50),
   last_name VARCHAR(50),
   salary INT
);

INSERT INTO employees VALUES
(1, 'John', 'Doe', 50000),
(2, 'Jane', 'Smith', 60000),
(3, 'Bob', 'Johnson', 55000),
(4, 'Alice', 'Williams', 70000);
```

A) Viewing all databases:-

```
SHOW DATABASES;
```

B) Viewing all tables in a database:-

```
USE Company;
SHOW TABLES;
```

C) Updating Records in a table:-

```
UPDATE employees
SET salary = 65000
WHERE employee_id = 2;
```

D) Deleting Records in a table:-

```
DELETE FROM employees
WHERE employee_id = 4;
```

4. To Perform the following SQL query on database: a. Altering a Table, Dropping/Truncating/Renaming Tables, Backing up / Restoring a Database

```
-- Add a new column to an existing table
ALTER TABLE your_table_name
ADD COLUMN new_column_name datatype;

-- Modify an existing column
ALTER TABLE your_table_name
MODIFY COLUMN existing_column_name new_datatype;

-- Drop a table
DROP TABLE your_table_name;

-- Truncate a table (remove all rows, but keep the table
structure)
TRUNCATE TABLE your_table_name;

-- Rename a table
RENAME TABLE old_table_name TO new_table_name;

-- Backup
mysqldump -u your_username -p your_database_name >
backup.sql

-- Restore
mysql -u your_username -p your_database_name <
backup.sql
```

5.For a given set of relation schemes, create tables and perform the following Simple Queries: Simple Queries with Aggregate functions, Queries with Aggregate functions (group by and having clause), Queries involving- Date Functions, String Functions , Math Functions

Aggregate functions:-

i) Sum ii) Avg iii) count

```
SELECT SUM(column_name) AS total_sum
FROM your_table_name;
SELECT AVG(column_name) AS average_value
FROM your_table_name;
SELECT COUNT(*) AS row_count
FROM your_table_name;
```

Queries with Aggregate Functions (GROUP BY and HAVING Clause):

```
SELECT category, SUM(quantity) AS total_quantity
FROM your_table_name
GROUP BY category;

SELECT department, AVG(salary) AS average_salary
FROM employee_table
GROUP BY department
HAVING AVG(salary) > 50000;
```

Queries involving Date Functions:

```
SELECT column_name, YEAR(date_column) AS
extracted_year
FROM your_table_name;

SELECT name, TIMESTAMPDIFF(YEAR, birthdate, CURDATE())
AS age
FROM person_table;
```

Queries involving String Functions:

```
SELECT CONCAT(first_name, ' ', last_name) AS full_name
FROM employee_table;

SELECT UPPER(column_name) AS uppercase_value,
LOWER(column_name) AS lowercase_value
FROM your_table_name;
```

Queries involving Math Functions:

```
SELECT column_name, SQRT(numeric_column) AS
square_root_value
FROM your_table_name;

SELECT column_name, ROUND(decimal_column) AS
rounded_value
FROM your_table_name;
```

6. To perform SQL query that demonstrate Join Queries- Inner Join, Outer Join, Left join, Right Join

```
SELECT employees.employee_id, employees.employee_name,
departments.department_name
FROM employees
INNER JOIN departments ON employees.department_id =
departments.department_id;

SELECT employees.employee_id, employees.employee_name,
departments.department_name
FROM employees
LEFT JOIN departments ON employees.department_id =
departments.department_id;

SELECT employees.employee_id, employees.employee_name,
departments.department_name
FROM employees
RIGHT JOIN departments ON employees.department_id =
departments.department_id;

SELECT employees.employee_id, employees.employee_name,
departments.department_name
FROM employees
FULL OUTER JOIN departments ON employees.department_id
= departments.department_id;
```

7.To perform SQL query that demonstrate following: Search conditions, Summary queries, Sub- queries, Subqueries- With IN clause, With EXISTS clause

Students Table:

student_id	student_name	age	grade
1	Alice	20	A
2	Bob	22	B
3	Charlie	21	A

Courses Table:

course_id	course_name	credits
101	Math	3
102	History	4
103	English	3

1. Search Conditions:

```
-- Find students who are 21 years old
SELECT * FROM students
WHERE age = 21;
```

2. Summary Queries:

```
-- Find the average age of students
SELECT AVG(age) AS average_age
FROM students;
```

3. Sub-queries:

```
-- Find students enrolled in courses with more than 3 credits
SELECT student_name
FROM students
WHERE student_id IN (
   SELECT student_id
   FROM enrollments
```

```
WHERE course_id IN (
      SELECT course_id
      FROM courses
      WHERE credits > 3
   )
);
```

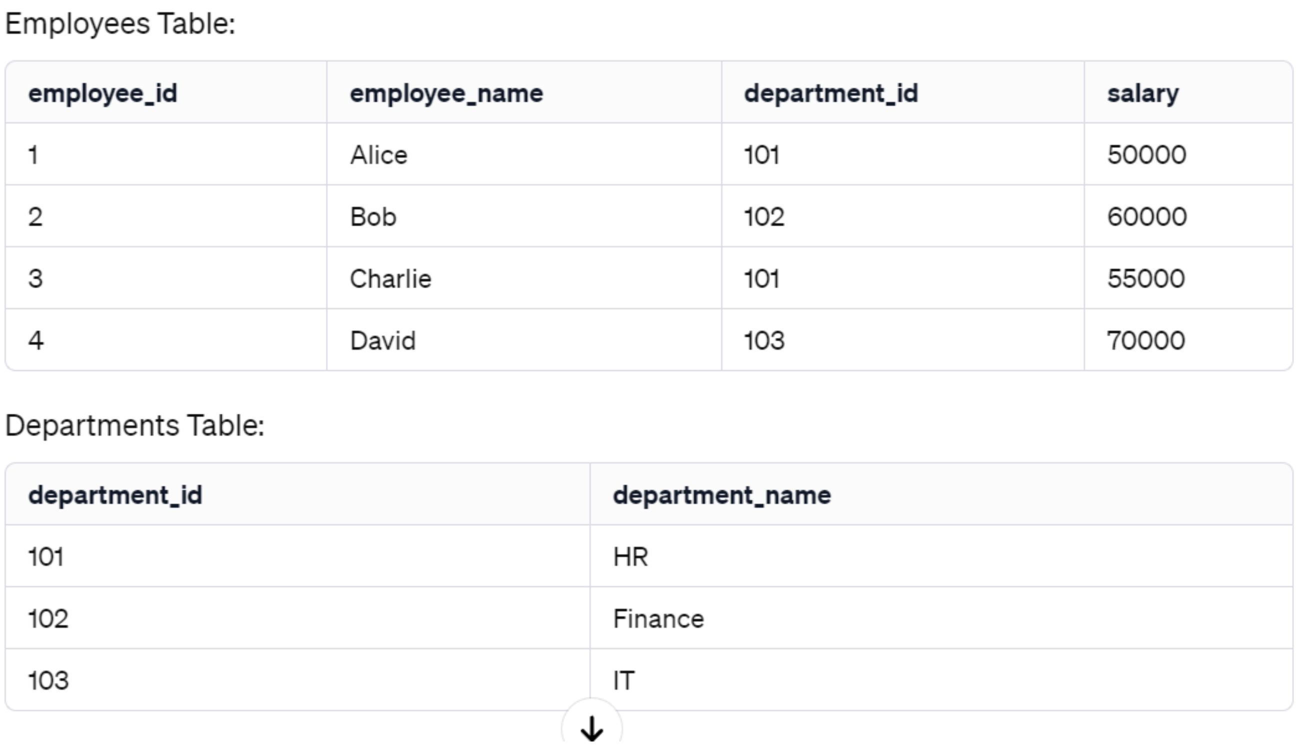
4. Subqueries with IN Clause:

```
-- Find students who have taken courses in English
SELECT student_name
FROM students
WHERE student_id IN (
    SELECT student_id
    FROM enrollments
    WHERE course_id IN (
        SELECT course_id
        FROM courses
        WHERE course_name = 'English'
    )
);
```

5. Subqueries with EXISTS Clause:

```
-- Find students who have enrolled in courses
SELECT student_name
FROM students s
WHERE EXISTS (
    SELECT 1
    FROM enrollments e
    WHERE e.student_id = s.student_id
);
```

8.To perform SQL query for extracting data from more than one table using SQL concept



Employees Table:

employee_id	employee_name	department_id	salary
1	Alice	101	50000
2	Bob	102	60000
3	Charlie	101	55000
4	David	103	70000

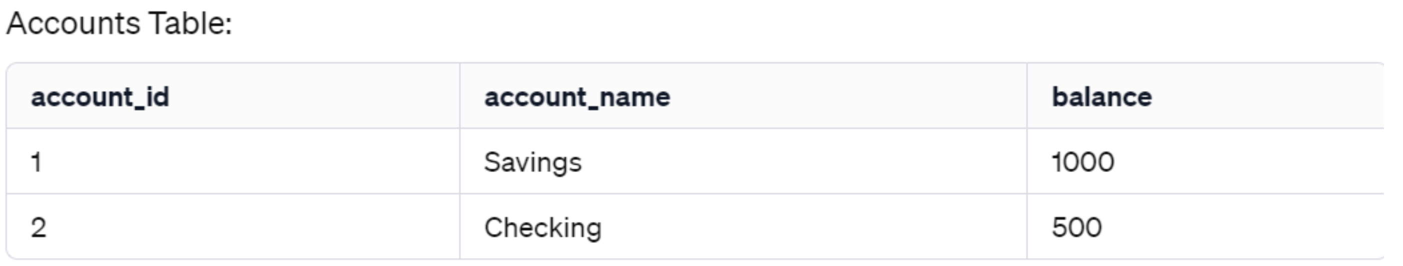
Departments Table:

department_id	department_name
101	HR
102	Finance
103	IT

```
-- Retrieve employee information along with their
 department names

SELECT e.employee_id, e.employee_name, e.salary,
d.department_name
FROM employees e
INNER JOIN departments d ON e.department_id =
d.department_id;
```

9.To perform SQL query to understand the concepts: Transaction, ROLL BACK, COMMIT & CHECK POINTS


Accounts Table:

account_id	account_name	balance
1	Savings	1000
2	Checking	500

```
-- Start a Transaction
BEGIN TRANSACTION;

-- Deduct amount from Savings Account (Account ID: 1)
UPDATE accounts
SET balance = balance - 200
WHERE account_id = 1;

-- Add the same amount to Checking Account (Account ID: 2)
UPDATE accounts
SET balance = balance + 200
WHERE account_id = 2;

-- Check the intermediate state of the accounts (optional)
SELECT * FROM accounts;

-- If everything is fine, commit the transaction
COMMIT;

-- If there's an issue, rollback the transaction
-- ROLLBACK;
```