# Scenario Based Questions

**SCENARIO 1 — Students & Courses**

**Tables**

```
CREATE TABLE Students (
   student_id INT PRIMARY KEY,
   student_name VARCHAR(50),
   city VARCHAR(50)
);

CREATE TABLE Courses (
   course_id INT PRIMARY KEY,
   course_name VARCHAR(50),
   fees INT
);

CREATE TABLE Enrollments (
   student_id INT,
   course_id INT,
   enrollment_date DATE,
   PRIMARY KEY(student_id, course_id),
   FOREIGN KEY(student_id) REFERENCES Students(student_id),
   FOREIGN KEY(course_id) REFERENCES Courses(course_id)
);
```

**Insert Data**

```
INSERT INTO Students VALUES
(1, 'Amit', 'Delhi'),
(2, 'Priya', 'Mumbai'),
(3, 'Raj', 'Delhi'),
```

(4, 'Sara', 'Pune'),

(5, 'John', 'Mumbai');

INSERT INTO Courses VALUES

(101, 'SQL', 5000),

(102, 'Python', 7000),

(103, 'Power BI', 6000);

INSERT INTO Enrollments VALUES

(1, 101, '2024-01-01'),

(1, 102, '2024-02-02'),

(2, 101, '2024-03-03'),

(3, 103, '2024-04-04'),

(5, 102, '2024-05-05');

**Questions & Answers**

**Q1. List all students with their enrolled courses**

**Q2. Show students who are not enrolled in any course (LEFT JOIN)**

**Q3. Show number of student enrollments per course**

**Q4. Get all students from Mumbai who enrolled in Python**

**Q5. Show course names even if no one enrolled (Right join alternative using left)**

**Q6. Show total fees collected from enrollments**

**SCENARIO 2 — Employees & Departments**

**Tables**

```
CREATE TABLE Departments (
    dept_id INT PRIMARY KEY,
    dept_name VARCHAR(50)
);
```

```
CREATE TABLE Employees (
    emp_id INT PRIMARY KEY,
    emp_name VARCHAR(50),
    dept_id INT,
    salary INT,
    FOREIGN KEY(dept_id) REFERENCES Departments(dept_id)
);
```

**Insert Data**

```
INSERT INTO Departments VALUES
(10, 'HR'),
(20, 'IT'),
(30, 'Finance');

INSERT INTO Employees VALUES
(1, 'Ravi', 10, 40000),
(2, 'Anu', 20, 55000),
(3, 'Sam', 20, 50000),
(4, 'Neha', 30, 60000),
(5, 'Tom', NULL, 45000);
```

**Questions & Answers**

**Q1. Show employees with department names**

**Q2. Employees working in IT department**

**Q3. Departments with no employees**

**Q4. Total salary paid per department**

**Q5. Show employees without a department**

**Q6. Highest salary employee with department**

**SCENARIO 3 — Orders & Customers**

**Tables**

```
CREATE TABLE Customers (
    cust_id INT PRIMARY KEY,
    cust_name VARCHAR(50),
    city VARCHAR(50)
);

CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    cust_id INT,
    order_amount INT,
    order_date DATE,
    FOREIGN KEY(cust_id) REFERENCES Customers(cust_id)
);
```

**Insert Data**

```
INSERT INTO Customers VALUES
(1, 'Arun', 'Delhi'),
(2, 'Kiran', 'Mumbai'),
(3, 'Faisal', 'Delhi'),
(4, 'Zara', 'Kolkata');

INSERT INTO Orders VALUES
(101, 1, 5000, '2024-01-01'),
(102, 1, 3000, '2024-01-05'),
(103, 2, 7000, '2024-01-10'),
(104, 3, 2000, '2024-01-20');
```

**Questions & Answers**

**Q1. List customer names with order amount**

**Q2. Show customers with no orders**

**Q3. Total orders made by each customer**

**Q4. Maximum order amount with customer**

**Q5. Show total order amount by city**

**Q6. Show customers who ordered more than 4000**

**SCENARIO 4 — Products & Sales**

**Tables**

```
CREATE TABLE Products (
   product_id INT PRIMARY KEY,
   product_name VARCHAR(50),
   price INT
);

CREATE TABLE Sales (
   sale_id INT PRIMARY KEY,
   product_id INT,
   qty INT,
   sale_date DATE,
   FOREIGN KEY(product_id) REFERENCES Products(product_id)
);
```

**Insert Data**

```
INSERT INTO Products VALUES
(100, 'Laptop', 50000),
(101, 'Mouse', 500),
(102, 'Keyboard', 1500),
(103, 'Monitor', 8000);

INSERT INTO Sales VALUES
```

(1, 100, 2, '2024-01-01'),

(2, 101, 5, '2024-01-02'),

(3, 102, 1, '2024-01-03'),

(4, 100, 1, '2024-01-05');

**Questions & Answers**

**Q1. Show product name & quantity sold**

**Q2. Show products never sold**

**Q3. Show total sales amount per product**

**Q4. Show highest selling product (qty)**

**Q5. Show total revenue**

**Q6. Show products sold more than 1 unit**

**SCENARIO 5 — Movies & Ratings**

**Tables**

```
CREATE TABLE Movies (
    movie_id INT PRIMARY KEY,
    movie_name VARCHAR(50),
    year INT
);

CREATE TABLE Ratings (
    rating_id INT PRIMARY KEY,
    movie_id INT,
    rating INT,
    FOREIGN KEY(movie_id) REFERENCES Movies(movie_id)
);
```

**Insert Data**

```
INSERT INTO Movies VALUES
```

(1, 'Inception', 2010),

(2, 'Avengers', 2012),

(3, 'Interstellar', 2014),

(4, 'Sholay', 1975);

INSERT INTO Ratings VALUES

(101, 1, 9),

(102, 1, 8),

(103, 2, 9),

(104, 3, 10);

**Questions & Answers**

**Q1. List movies with ratings**

**Q2. List movies which have no ratings**

**Q3. Show average rating of each movie**

**Q4. Highest rated movie**

**Q5. Movies released after 2010 with ratings**

**Q6. Show how many ratings each movie received**