SQL Placement Preparation Notes

1. SQL Basics

- **SQL**: Structured Query Language, used for managing relational databases.
- **Types of SQL commands**:
  - **DDL**: CREATE, ALTER, DROP
  - **DML**: SELECT, INSERT, UPDATE, DELETE
  - **DCL**: GRANT, REVOKE
  - **TCL**: COMMIT, ROLLBACK, SAVEPOINT

2. Data Types

- **INT**, **VARCHAR(n)**, **CHAR(n)**, **DATE**, **FLOAT**, **BOOLEAN**

3. Constraints

- **NOT NULL**: No empty values
- **UNIQUE**: No duplicate values
- **PRIMARY KEY**: NOT NULL + UNIQUE
- **FOREIGN KEY**: References another table
- **CHECK**: Ensures value condition
- **DEFAULT**: Sets default value

4. Joins

- **INNER JOIN**: Common rows only
- **LEFT JOIN**: All from left + matched right
- **RIGHT JOIN**: All from right + matched left
- **FULL OUTER JOIN**: All rows from both
- **SELF JOIN**: Join table with itself

5. Subqueries & CTEs

- **Subquery**: Query inside another query
- **CTE (WITH)**: Temporary result set for reuse

6. GROUP BY & Aggregates

- Used with functions: COUNT, SUM, AVG, MAX, MIN
- **HAVING**: Filter after GROUP BY

7. Set Operations

- **UNION**: Combines, removes duplicates
- **UNION ALL**: Combines, keeps duplicates
- **INTERSECT**: Common rows
- **EXCEPT**: Rows in first not in second

8. Indexes

- Improve query speed
- **CREATE INDEX index_name ON table(column);**

9. Normalization

- Removes redundancy, increases integrity
- **1NF (First Normal Form)**: No repeating groups or arrays. All attributes must contain atomic (indivisible) values.
- **2NF (Second Normal Form)**: 1NF + every non-key attribute fully functionally dependent on the entire primary key (eliminates partial dependencies).
- **3NF (Third Normal Form)**: 2NF + no transitive dependency (non-key attribute should not depend on another non-key attribute).
- **BCNF (Boyce-Codd Normal Form)**: A stronger version of 3NF. Every determinant must be a candidate key.
- **4NF (Fourth Normal Form)**: BCNF + no multi-valued dependencies (an attribute should not have multiple independent values).

- **5NF (Fifth Normal Form)**: 4NF + no join dependency. Data should be reconstructable from smaller relations without redundancy.

10. Transactions

- **ACID** properties:
  - **Atomicity**: All or nothing
  - **Consistency**: Valid state only
  - **Isolation**: Transactions don't interfere
  - **Durability**: Changes persist after commit

Frequently Asked Interview Questions & Answers

1. **Difference between WHERE and HAVING?**
   - `WHERE` filters rows before aggregation.
   - `HAVING` filters after aggregation (used with GROUP BY).
2. **What is the difference between DELETE, TRUNCATE, and DROP?**
   - `DELETE` : Removes rows, can be rolled back.
   - `TRUNCATE` : Removes all rows, faster, cannot be rolled back.
   - `DROP` : Deletes table structure permanently.
3. **Explain different types of JOINs with examples.**
   - `INNER JOIN` : Matches rows from both tables.
   - `LEFT JOIN` : All left rows + matched right.
   - `RIGHT JOIN` : All right rows + matched left.
   - `FULL JOIN` : All rows from both tables.
   - Example: `SELECT * FROM A LEFT JOIN B ON A.id = B.id;`
4. **What is a PRIMARY KEY vs FOREIGN KEY?**
   - `PRIMARY KEY` : Uniquely identifies each row in a table.
   - `FOREIGN KEY` : Enforces link between two tables.
5. **How does indexing improve performance?**
   - Indexes help locate data quickly, avoiding full table scans.
   - Downside: Slower INSERT/UPDATE due to maintenance.
6. **Write a query to find the 2nd highest salary.**

```
SELECT MAX(salary) FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

7. **What are window functions?**
    - Functions like `ROW_NUMBER()`, `RANK()`, `DENSE_RANK()` that work over a window of rows.
    - Example:

```
SELECT name, salary, RANK() OVER (ORDER BY salary DESC) FROM employees;
```

8. **What is normalization and its types?**
    - Technique to reduce data redundancy and improve data integrity.
    - Types:
        - 1NF (First Normal Form): No repeating groups or arrays. All attributes must contain atomic (indivisible) values.
        - 2NF (Second Normal Form): 1NF + every non-key attribute fully functionally dependent on the entire primary key (eliminates partial dependencies).
        - 3NF (Third Normal Form): 2NF + no transitive dependency (non-key attribute should not depend on another non-key attribute).
        - BCNF (Boyce-Codd Normal Form): A stronger version of 3NF. Every determinant must be a candidate key.
        - 4NF (Fourth Normal Form): BCNF + no multi-valued dependencies (an attribute should not have multiple independent values).
        - 5NF (Fifth Normal Form): 4NF + no join dependency. Data should be reconstructable from smaller relations without redundancy.
9. **Difference between UNION and UNION ALL?**
    - `UNION` : Removes duplicates.
    - `UNION ALL` : Keeps all duplicates.
10. **Explain ACID properties with an example.**

- **Atomicity**: All operations succeed or fail.
- **Consistency**: Always valid state (e.g., no negative bank balance).
- **Isolation**: Transactions don't conflict (e.g., booking same seat).
- **Durability**: Changes persist even after power loss.

Practice Queries

```sql
-- 1. Second highest salary
SELECT MAX(salary) FROM employees WHERE salary < (SELECT MAX(salary) FROM employees);

-- 2. Employees with duplicate names
SELECT name, COUNT(*) FROM employees GROUP BY name HAVING COUNT(*) > 1;

-- 3. Employees in multiple departments
SELECT emp_id FROM employee_dept GROUP BY emp_id HAVING COUNT(DISTINCT dept_id) > 1;

-- 4. Join employees and departments
SELECT e.name, d.dept_name FROM employees e JOIN departments d ON e.dept_id = d.id;
```