

Cheat Sheet for comprehensive CIW Database Specialist

SQL Basics

Data Types

- **Numeric:** INT, FLOAT, DECIMAL(p,s)
- **Character:** CHAR(n), VARCHAR(n), TEXT
- **Date/Time:** DATE, TIME, DATETIME, TIMESTAMP
- **Boolean:** BOOLEAN (TRUE/FALSE)

Operators

- **Arithmetic:** +, -, *, /, %
- **Comparison:** =, <>, >, <, >=, <=
- **Logical:** AND, OR, NOT

Comments

- **Single-line:** `-- This is a comment``
- **Multi-line:** `/* This is a multi-line comment */``

SQL Commands

Data Definition Language (DDL)

- **CREATE TABLE:** `CREATE TABLE table_name (column1 datatype, column2 datatype, ...);``
- **ALTER TABLE:** `ALTER TABLE table_name ADD column_name datatype;``
- **DROP TABLE:** `DROP TABLE table_name;``
- **TRUNCATE TABLE:** `TRUNCATE TABLE table_name;``

Data Manipulation Language (DML)

- **INSERT:** `INSERT INTO table_name (column1, column2) VALUES (value1, value2);``
- **UPDATE:** `UPDATE table_name SET column1 = value1 WHERE condition;``
- **DELETE:** `DELETE FROM table_name WHERE condition;``

Data Query Language (DQL)

- **SELECT:** `SELECT column1, column2 FROM table_name WHERE condition;``

- **SELECT DISTINCT:** `SELECT DISTINCT column1 FROM table\_name;`
- **ORDER BY:** `SELECT column1 FROM table\_name ORDER BY column1 ASC|DESC;`
- **GROUP BY:** `SELECT column1, COUNT(\*) FROM table\_name GROUP BY column1;`
- **HAVING:** `SELECT column1, COUNT(\*) FROM table\_name GROUP BY column1 HAVING COUNT(\*) > 1;`

Data Control Language (DCL)

- **GRANT:** `GRANT permission ON object TO user;`
- **REVOKE:** `REVOKE permission ON object FROM user;`

Joins

Types of Joins

- **INNER JOIN:** `SELECT \* FROM table1 INNER JOIN table2 ON table1.column = table2.column;`
- **LEFT JOIN:** `SELECT \* FROM table1 LEFT JOIN table2 ON table1.column = table2.column;`
- **RIGHT JOIN:** `SELECT \* FROM table1 RIGHT JOIN table2 ON table1.column = table2.column;`
- **FULL OUTER JOIN:** `SELECT \* FROM table1 FULL OUTER JOIN table2 ON table1.column = table2.column;`
- **CROSS JOIN:** `SELECT \* FROM table1 CROSS JOIN table2;`

Subqueries

- **Single-row Subquery:** `SELECT column1 FROM table1 WHERE column2 = (SELECT column2 FROM table2 WHERE condition);`
- **Multi-row Subquery:** `SELECT column1 FROM table1 WHERE column2 IN (SELECT column2 FROM table2 WHERE condition);`

Functions

Aggregate Functions

- **COUNT:** `SELECT COUNT(column1) FROM table\_name;`
- **SUM:** `SELECT SUM(column1) FROM table\_name;`
- **AVG:** `SELECT AVG(column1) FROM table\_name;`

- **MIN:** `SELECT MIN(column1) FROM table\_name;`
- **MAX:** `SELECT MAX(column1) FROM table\_name;`

String Functions

- **CONCAT:** `SELECT CONCAT(column1, ' ', column2) FROM table\_name;`
- **SUBSTRING:** `SELECT SUBSTRING(column1, start, length) FROM table\_name;`
- **LENGTH:** `SELECT LENGTH(column1) FROM table\_name;`

Date Functions

- **NOW:** `SELECT NOW();`
- **DATE_FORMAT:** `SELECT DATE\_FORMAT(column1, '%Y-%m-%d') FROM table\_name;`
- **DATEDIFF:** `SELECT DATEDIFF(column1, column2) FROM table\_name;`

Indexes

- **CREATE INDEX:** `CREATE INDEX index\_name ON table\_name (column1);`
- **DROP INDEX:** `DROP INDEX index\_name ON table\_name;`

Transactions

- **BEGIN TRANSACTION:** `BEGIN TRANSACTION;`
- **COMMIT:** `COMMIT;`
- **ROLLBACK:** `ROLLBACK;`

Views

- **CREATE VIEW:** `CREATE VIEW view\_name AS SELECT column1, column2 FROM table\_name WHERE condition;`
- **DROP VIEW:** `DROP VIEW view\_name;`

Stored Procedures

- **CREATE PROCEDURE:** `CREATE PROCEDURE procedure\_name AS BEGIN SQL\_statements END;`
- **EXECUTE PROCEDURE:** `EXEC procedure\_name;`

Triggers

- **CREATE TRIGGER:** `CREATE TRIGGER trigger\_name BEFORE INSERT ON table\_name FOR EACH ROW BEGIN SQL\_statements END;`
- **DROP TRIGGER:** `DROP TRIGGER trigger\_name;`

Tips and Tricks

- **Use EXPLAIN:** `EXPLAIN SELECT \* FROM table\_name;` to analyze query performance.
- **Normalization:** Follow normalization rules (1NF, 2NF, 3NF) to avoid data redundancy.
- **Backup:** Regularly back up your database using `mysqldump` or similar tools.
- **Performance Tuning:** Use indexes judiciously, avoid using `SELECT \*`, and optimize joins.

Examples

Example 1: Creating a Table

```
CREATE TABLE Employees (  
    EmployeeID INT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    HireDate DATE  
);
```

Example 2: Inserting Data

```
INSERT INTO Employees (EmployeeID, FirstName, LastName, HireDate)  
VALUES (1, 'John', 'Doe', '2023-01-15');
```

Example 3: Querying Data

```
SELECT FirstName, LastName FROM Employees WHERE HireDate > '2023-01-01';
```

Example 4: Updating Data

```
UPDATE Employees SET HireDate = '2023-01-20' WHERE EmployeeID = 1;
```

Example 5: Deleting Data

```
DELETE FROM Employees WHERE EmployeeID = 1;
```

Conclusion

This cheat sheet provides a comprehensive overview of essential SQL commands, functions, and best practices for a CIW Database Specialist. Use these commands and tips to efficiently manage and query databases.

By Ahmed Baheeg Khorshid

ver 1.0