SQL Cheat Sheet

Anil pankaj

Querying Data in SQL

SELECT

Retrieve Data From One Or More Tables

```
SELECT * FROM employees;
```

DISTINCT

Select Unique Values From A Column

```
SELECT DISTINCT department FROM employees;
```

WHERE

Filter Rows Based On Specified Conditions

```
SELECT * FROM employees WHERE salary > 55000.00;
```

LIMIT

Limit The Number Of Rows Returned In The Result Set

```
SELECT * FROM employees LIMIT 3;
```

FETCH

Retrieve A Specified Number Of Rows From The Result Set

```
SELECT * FROM employees FETCH FIRST 3 ROWS ONLY;
```

Filtering Data in SQL

WHERE

Filter Rows Based On Specified Conditions

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

LIKE

Match A Pattern In A Column

```
SELECT * FROM employees WHERE first_name LIKE 'JX';
```

IN

Match Any Value In A List

```
SELECT * FROM employees WHERE department IN ('HR', 'Finance');
```

BETWEEN

Match Values Within A Specified Range

```
SELECT * FROM employees WHERE salary BETWEEN 50000 AND 60000;
```

IS NULL

Match NULL Values

```
SELECT * FROM employees WHERE department IS NULL;
```

SQL Operator

AND

Combines Multiple Conditions In A WHERE Clause

```
SELECT * FROM employees WHERE department = 'IT' AND salary > 60000;
```

OR

Specifies Multiple Conditions Where Any One Of Them Should Be True

```
SELECT * FROM employees WHERE department = 'HR' OR department = 'Finance';
```

NOT

Negates A Condition

```
SELECT * FROM employees WHERE NOT department = 'IT';
```

ORDER BY

Sorts the Result Set in Ascending or Descending Order

```
SELECT * FROM employees ORDER BY salary DESC;
```

GROUP BY

Groups Rows that have the Same Values into Summary Rows

```
SELECT department, COUNT(*) AS employee_count FROM employees GROUP BY department;
```

Aggregation Data in SQL

COUNT

Count The Number Of Rows In A Result Set

```
SELECT COUNT(*) FROM employees;
```

SUM

Calculate The Sum Of Values In A Column

```
SELECT SUM(salary) FROM employees;
```

AVG

Calculate The Average Value Of A Column

```
SELECT AVG(salary) FROM employees;
```

MIN

Find the Minimum Value in a Column

```
SELECT MIN(salary) FROM employees;
```

MAX

Find the Maximum Value in a Column

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

Joins in SQL

INNER JOIN

Retrieves Records That Have Matching Values in Both Tables

```
SELECT * FROM employees INNER JOIN departments ON employees.department_id = departments.department_id;
```

LEFT JOIN

Retrieves All Records from the Left Table and the Matched Records from the Right Table

```
SELECT * FROM employees LEFT JOIN departments ON employees.department_id = departments.department_id;
```

RIGHT JOIN

Retrieves All Records from the Right Table and the Matched Records from the Left Table

```
SELECT * FROM employees RIGHT JOIN departments ON employees.department_id = departments.department_id;
```

FULL OUTER JOIN

Retrieves All Records When There Is a Match in Either the Left or Right Table

```
SELECT * FROM employees FULL OUTER JOIN departments ON employees.department_id = departments.department_id;
```

CROSS JOIN

Retrieves the Cartesian Product of the Two Tables

```
SELECT * FROM employees CROSS JOIN departments;
```

Indexes & Transactions in SQL

CREATE INDEX

Create an Index on a Table

```
CREATE INDEX idx_department ON employees (department);
```

DROP INDEX

Remove an Index

```
DROP INDEX IF EXISTS idx_department;
```

BEGIN TRANSACTION

Start a New Transaction

```
BEGIN TRANSACTION;
```

COMMIT

Save Changes Made During the Current Transaction

```
COMMIT;
```

ROLLBACK

Undo Changes Made During the Current Transaction

```
ROLLBACK;
```

To Learn More Commands, You can read this article here.