SQL MASTERY CHEATSHEET

SQL Command Categories

DDL (Data Definition Language)

CREATE

```
CREATE TABLE
table_name (
  column1 datatype1,
  column2 datatype2
);
```

ALTER

```
ALTER TABLE table_name
ADD column_name
datatype;
```

DROP

```
DROP TABLE table_name;
```

TRUNCATE

```
TRUNCATE TABLE
table_name;
```

DML (Data Manipulation Language)

INSERT

```
INSERT INTO table_name
VALUES (value1,
value2);
```

UPDATE

```
UPDATE table_name
SET column1 = value1
WHERE condition;
```

DELETE

```
DELETE FROM table_name
WHERE condition;
```

DCL & TCL

Command	Syntax
GRANT	`GRANT privilege ON object TO user;`
REVOKE	`REVOKE privilege ON object FROM user;`
COMMIT	`COMMIT;`
ROLLBACK	`ROLLBACK [TO savepoint_name];`

SQL Data Types

String Types

Type	Description	Syntax
CHAR	Fixed length	`column_name CHAR(size)`
VARCHAR	Variable length	`column_name VARCHAR(max_size)`
TEXT	Large text	`column_name TEXT`

Numeric Types

Type	Description	Syntax
INTEGER	Whole numbers	`column_name INTEGER`
DECIMAL	Exact decimal	`column_name DECIMAL(p,s)`
FLOAT	Approximate decimal	`column_name FLOAT`

Date/Time Types

Type	Format	Syntax
DATE	YYYY-MM-DD	`column_name DATE`
TIME	HH:MM:SS	`column_name TIME`
TIMESTAMP	YYYY-MM-DD HH:MM:SS	`column_name TIMESTAMP`

SQL Constraints

PRIMARY KEY

```
CREATE TABLE
table_name (
  id INT PRIMARY
KEY,
  -- or --
  CONSTRAINT pk_name
PRIMARY KEY
(column1, column2)
);
```

FOREIGN KEY

```
CREATE TABLE
table_name (
  order_id INT,
  FOREIGN KEY
(order_id)
    REFERENCES
orders(id)
);
```

CHECK

```
CREATE TABLE
table_name (
  age INT CHECK (age
>= 18),
  -- or --
  CONSTRAINT
chk_person CHECK
(age >= 18 AND city
= 'NY')
);
```

UNIQUE

```
CREATE TABLE
table_name (
  email VARCHAR(100)
UNIQUE,
  -- or --
  CONSTRAINT
unq_name UNIQUE
(column1, column2)
);
```

Basic Queries and Operators

SELECT Statement Components

```
SELECT [DISTINCT] column1, column2, aggregate_function(column3)
FROM table_name
[JOIN join_table ON join_condition]
[WHERE condition]
[GROUP BY column1, column2]
[HAVING group_condition]
[ORDER BY column1 [ASC|DESC]]
[LIMIT n [OFFSET m]];
```

Arithmetic Operators

`+, -, *, /, %`

Comparison Operators

`=, >, <, >=, <=, <>, !=, !>, !<`

Logical Operators

`AND, OR, NOT, IN, BETWEEN, LIKE, IS NULL`

SQL JOINs

INNER JOIN

```
SELECT *
FROM table1 t1
INNER JOIN table2 t2
  ON t1.id = t2.id;
```

Venn Notation: A ∩ B

LEFT JOIN

```
SELECT *
FROM table1 t1
LEFT JOIN table2 t2
  ON t1.id = t2.id;
```

Venn Notation: A - B ∪ A ∩ B

RIGHT JOIN

```
SELECT *
FROM table1 t1
RIGHT JOIN table2 t2
  ON t1.id = t2.id;
```

Venn Notation: B - A ∪ A ∩ B

FULL OUTER JOIN

```
SELECT *
FROM table1 t1
FULL OUTER JOIN table2 t2
  ON t1.id = t2.id;
```

Venn Notation: A ∪ B

CROSS JOIN

```
SELECT *
FROM table1
CROSS JOIN table2;
```

Venn Notation: A × B (Cartesian Product)

NATURAL JOIN

```
SELECT *
FROM table1 t1
NATURAL JOIN table2 t2;
```

Venn Notation: A ∩ B (based on matching columns)

EXCLUSIVE JOIN

```
SELECT *
FROM table1 t1
EXCLUSIVE JOIN table2 t2
  ON t1.id = t2.id;
```

Venn Notation: (A - B) ∪ (B - A)

Window Functions

Basic Syntax

```
function_name() OVER (
  [PARTITION BY column1]
  [ORDER BY column2]
  [frame_clause]
)
```

Ranking Functions

```
SELECT
  ROW_NUMBER() OVER (ORDER BY column),
  RANK() OVER (ORDER BY column),
  DENSE_RANK() OVER (ORDER BY column)
FROM table_name;
```

Analytic Functions

```
SELECT
  LAG(column) OVER (ORDER BY date),
  LEAD(column) OVER (ORDER BY date),
  FIRST_VALUE(column) OVER (
    ORDER BY date
    ROWS BETWEEN UNBOUNDED PRECEDING
    AND CURRENT ROW
  )
FROM table_name;
```

CTEs and Subqueries

Common Table Expression (CTE)

```
WITH cte_name AS (
  SELECT column1, column2
  FROM table_name
  WHERE condition
)
SELECT *
FROM cte_name;
```

Recursive CTE

```
WITH RECURSIVE cte_name AS (
  -- Base case
  SELECT column1, column2
  FROM table_name
  WHERE condition

  UNION ALL

  -- Recursive case
  SELECT t.column1, t.column2
  FROM table_name t
  INNER JOIN cte_name c
    ON t.id = c.parent_id
)
SELECT * FROM cte_name;
```