IBM Db2 Cheatsheet

A quick reference guide for IBM Db2, covering essential commands, data types, SQL syntax, and administrative tasks.

Basic SQL Commands

Data Definition Language (DDL)

`CREATE TABLE`	Creates a new table.
	Example:
	``sql
	CREATE TABLE employees (
	  id INT PRIMARY KEY,
	  name VARCHAR(255),
	  salary DECIMAL(10, 2)
	);
	``

`ALTER TABLE`	Modifies an existing table.
	Example:
	``sql
	ALTER TABLE employees
	ADD COLUMN department VARCHAR(255);
	``

`DROP TABLE`	Deletes a table.
	Example:
	``sql
	DROP TABLE employees;
	``

`CREATE INDEX`	Creates an index on a table.
	Example:
	``sql
	CREATE INDEX idx_name ON employees (name);
	``

`DROP INDEX`	Deletes an index.
	Example:
	``sql
	DROP INDEX idx_name;
	``

`CREATE VIEW`	Creates a virtual table based on the result-set of an SQL statement.
	Example:
	``sql
	CREATE VIEW high_earners AS
	SELECT id, name FROM employees WHERE salary > 50000;
	``

Data Manipulation Language (DML)

`INSERT`	Inserts data into a table.
	Example:
	``sql
	INSERT INTO employees (id, name, salary)
	VALUES (1, 'John Doe', 60000);
	``

`UPDATE`	Updates existing data in a table.
	Example:
	``sql
	UPDATE employees
	SET salary = 65000
	WHERE id = 1;
	``

`DELETE`	Deletes data from a table.
	Example:
	``sql
	DELETE FROM employees
	WHERE id = 1;
	``

`SELECT`	Retrieves data from a table.
	Example:
	``sql
	SELECT id, name, salary
	FROM employees;
	``

`MERGE`	Performs insert, update, or delete operations based on a condition.
	Example:
	``sql
	MERGE INTO target_table AS T
	USING source_table AS S
	ON (T.key = S.key)
	WHEN MATCHED THEN
	  UPDATE SET T.column1 = S.column1
	WHEN NOT MATCHED THEN
	  INSERT (key, column1) VALUES (S.key, S.column1);
	``

Advanced SQL and Functions

Common Functions

`COUNT()`	Returns the number of rows. **Example:** `SELECT COUNT(*) FROM employees;`
`AVG()`	Returns the average value. **Example:** `SELECT AVG(salary) FROM employees;`
`SUM()`	Returns the sum of values. **Example:** `SELECT SUM(salary) FROM employees;`
`MIN()`	Returns the minimum value. **Example:** `SELECT MIN(salary) FROM employees;`
`MAX()`	Returns the maximum value. **Example:** `SELECT MAX(salary) FROM employees;`
`UPPER()`	Converts a string to uppercase. **Example:** `SELECT UPPER(name) FROM employees;`
`LOWER()`	Converts a string to lowercase. **Example:** `SELECT LOWER(name) FROM employees;`
`LENGTH()`	Returns the length of a string. **Example:** `SELECT LENGTH(name) FROM employees;`

Joins

`INNER JOIN`	Returns rows when there is a match in both tables.
	Example:
	`SELECT * FROM employees`
	`INNER JOIN departments`
	`ON employees.department_id = departments.id;`

`LEFT JOIN`	Returns all rows from the left table, and the matched rows from the right table.
	Example:
	`SELECT * FROM employees`
	`LEFT JOIN departments`
	`ON employees.department_id = departments.id;`

`RIGHT JOIN`	Returns all rows from the right table, and the matched rows from the left table.
	Example:
	`SELECT * FROM employees`
	`RIGHT JOIN departments`
	`ON employees.department_id = departments.id;`

`FULL OUTER JOIN`	Returns all rows when there is a match in one of the tables.
	Example:
	`SELECT * FROM employees`
	`FULL OUTER JOIN departments`
	`ON employees.department_id = departments.id;`

Db2 Administration

Basic Administration Commands

`db2start`	Starts the Db2 database manager. **Example:** `db2start`
`db2stop`	Stops the Db2 database manager. **Example:** `db2stop`
`db2 connect to <database_name>`	Connects to a specific database. **Example:** `db2 connect to sample`
`db2 disconnect <database_name>`	Disconnects from a database. **Example:** `db2 disconnect sample`
`db2 list databases`	Lists all databases known to the Db2 instance. **Example:** `db2 list databases`
`db2 backup database <database_name> to <backup_location>`	Backs up a database to a specified location. **Example:** `db2 backup database sample to /backup`
`db2 restore database <database_name> from <backup_location>`	Restores a database from a backup. **Example:** `db2 restore database sample from /backup`

User and Permissions

`CREATE USER`	Creates a new user (typically managed at the OS level). **Note:** Db2 relies on the operating system for user authentication.
`GRANT`	Grants privileges to a user or role. **Example:** `GRANT SELECT ON TABLE employees TO USER john;`
`REVOKE`	Revokes privileges from a user or role. **Example:** `REVOKE SELECT ON TABLE employees FROM USER john;`

Data Types

Common Data Types

`INTEGER` or `INT`	Stores integer values. **Example:** `id INT`
`SMALLINT`	Stores small integer values. **Example:** `age SMALLINT`
`BIGINT`	Stores large integer values. **Example:** `population BIGINT`
`DECIMAL(p, s)` or `NUMERIC(p, s)`	Stores exact numeric values with precision `p` and scale `s`. **Example:** `salary DECIMAL(10, 2)`
`REAL`	Stores single-precision floating-point numbers. **Example:** `temperature REAL`

`DOUBLE` or `DOUBLE PRECISION`	Stores double-precision floating-point numbers. **Example:** `avg_temp DOUBLE`
`VARCHAR(n)`	Stores variable-length character strings with a maximum length of `n`. **Example:** `name VARCHAR(255)`
`CHAR(n)`	Stores fixed-length character strings with a length of `n`. **Example:** `state_code CHAR(2)`
`DATE`	Stores dates. **Example:** `birth_date DATE`
`TIME`	Stores times. **Example:** `start_time TIME`
`TIMESTAMP`	Stores date and time values. **Example:** `created_at TIMESTAMP`
`BLOB`	Stores Binary Large Objects (BLOBs). **Example:** `image BLOB`
`CLOB`	Stores Character Large Objects (CLOBs). **Example:** `text_data CLOB`