


# PostgreSQL Printable Cheat Sheet and Viva Prep Guide

---

## 1. Getting Started with PostgreSQL


- **PostgreSQL:** Open-source, ACID-compliant, object-relational DBMS supporting advanced features like full-text search, custom functions, and JSON.
- **Install Methods:**
  - Native: OS-level installation.
  - Docker: `docker pull postgres`
- **CLI (`psql`) Essentials:**
  - `\l` – list DBs, `\c dbname` – connect, `\dt` – list tables, `\d tablename` – describe table, `\q` – quit
- **pgAdmin:**
  - GUI for database management and query execution.
  - Connects to Docker via localhost and exposed port (5432).

 **Viva Notes:** Docker volumes = persistence; pgAdmin is GUI, `psql` is scriptable; both are essential.

---

## 2. Introduction to Relational Databases

- **Tables:** Composed of rows (records) and columns (fields).
- **Data Types:**
  - Basic: `INTEGER`, `TEXT`, `BOOLEAN`, `DATE`, `NUMERIC`
  - Advanced: `UUID`, `JSON/JSONB`, `ARRAY`
- **Relationships:**
  - One-to-many via foreign keys
  - Many-to-many via junction tables
- **SQL Basics:** `INSERT`, `SELECT`, `UPDATE`, `DELETE`

 **Viva Notes:** `TEXT` ≠ length limit, `VARCHAR(n)` does; `UUID` > `SERIAL` for uniqueness; Joins implement relationships.

---


## 3. Building a Database

- **Schemas:** Logical namespace inside a database.
- **Primary Keys:** Unique row identifiers (use `SERIAL`, `UUID`).
- **Foreign Keys:** References between tables. Example:

SQL

```
FOREIGN KEY (dept_id) REFERENCES departments(id) ON DELETE  
CASCADE;
```


- **Referential Integrity:** Maintains consistency via `CASCADE`, `SET NULL`, `RESTRICT` actions.

 *Viva Notes:* Cascading deletes avoid orphan records; schemas help with modular/multi-tenant DBs.

---

## 4. Retrieving Data with Queries


- **CSV Import:** `COPY table FROM 'file.csv' DELIMITER ',' CSV HEADER;`
- **SELECT Queries:** Use `WHERE`, `ORDER BY`, `LIMIT`, `GROUP BY`, `HAVING`.
- **Joins:**
  - `INNER JOIN`: Matching rows only
  - `LEFT JOIN`: All left + matched right
  - `FULL JOIN`: All from both, unmatched as `NULL`
- **Views:**
  - Logical abstraction: `CREATE VIEW v_name AS SELECT...`
  - Use `MATERIALIZED VIEW` for performance gains

 *Viva Notes:* Views = abstraction layer; Joins reflect entity relationships; CSV import must match schema.

---

## 5. Managing Data


- **Indexes:**
  - Auto: On PK/Unique
  - Manual: `CREATE INDEX idx_name ON table(col);`
  - Types:
    - `BTREE` (default)
    - `GIN` for JSON/arrays
    - `BRIN` for ordered big data
- **Defaults:** `DEFAULT 0`, `DEFAULT now()`
- **Constraints:**
  - `NOT NULL`, `UNIQUE`, `CHECK`, `PRIMARY KEY`, `FOREIGN KEY`
  - Example: `CHECK (salary >= 0)`

 *Viva Notes:* Constraints enforce rules at DB level; `GIN` accelerates JSON search.


---

## 6. Database Administration

- **Roles & Permissions:**
  - Create: `CREATE ROLE username LOGIN PASSWORD 'pass';`
  - Grant: `GRANT SELECT ON table TO user;`
- **Backups:**
  - Logical: `pg_dump db > backup.sql`, restore with `psql`
  - Custom: `pg_dump -Fc`, restore with `pg_restore`
- **Server Control:**
  - `pg_ctl start`, `systemctl start postgresql`

 *Viva Notes:* `pg_dump` = SQL, `pg_basebackup` = physical; Superuser access should be limited.

---

 Use this cheat sheet to confidently prepare for PostgreSQL exams, viva rounds, and hands-on tasks. Need practice SQL queries or flashcards? Just ask!