

## POSTGRESQL COMMANDS – COMPLETE CHEAT SHEET

### 1. PostgreSQL Terminal Commands

- `psql -U postgres` – Login to PostgreSQL
- `\q` – Quit
- `\l` – List all databases
- `\c database_name` – Connect to DB
- `\dt` – List tables
- `\dn` – List schemas
- `\dv` – List views
- `\df` – List functions
- `\d table_name` – Describe table

### 2. Database Commands

`CREATE DATABASE school;`

`DROP DATABASE school;`

`ALTER DATABASE school RENAME TO college;`

### 3. Table Commands

```
CREATE TABLE students (  
id SERIAL PRIMARY KEY,  
name VARCHAR(50),  
age INT,  
city VARCHAR(50),  
marks INT  
);
```

### 4. Insert / Select / Update / Delete

```
INSERT INTO students(name, age, city, marks)  
VALUES ('Hari', 22, 'Chennai', 90);
```

```
SELECT * FROM students;
```

```
SELECT name, city FROM students;
```

```
SELECT * FROM students WHERE marks > 80;
```

```
UPDATE students SET city='Coimbatore' WHERE id=1;
```

```
DELETE FROM students WHERE id=3;
```

#### 5. Filtering

```
SELECT * FROM students WHERE city = 'Chennai';
```

```
SELECT * FROM students WHERE marks BETWEEN 70 AND 90;
```

```
SELECT * FROM students WHERE name LIKE 'H%';
```

#### 6. Sorting

```
SELECT * FROM students ORDER BY marks DESC;
```

#### 7. Aggregate Functions

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

```
SELECT AVG(marks) FROM students;
```

```
SELECT MIN(marks), MAX(marks) FROM students;
```

GROUP BY:

```
SELECT city, COUNT(*) FROM students GROUP BY city;
```

#### 8. Index Commands

```
CREATE INDEX idx_city ON students(city);
```

View indexes:

```
SELECT indexname, indexdef FROM pg_indexes WHERE tablename='students';
```

Drop index:

```
DROP INDEX idx_city;
```

#### 9. Trigger Commands

```
CREATE TABLE student_log (  
  id SERIAL PRIMARY KEY,  
  student_name VARCHAR(50),  
  action VARCHAR(10),  
  log_time TIMESTAMP DEFAULT NOW()  
);
```

Trigger Function:

```
CREATE OR REPLACE FUNCTION log_student_insert()  
RETURNS TRIGGER AS $$  
  
BEGIN  
  
  INSERT INTO student_log(student_name, action)  
  VALUES (NEW.name, 'INSERT');  
  
  RETURN NEW;  
  
END;  
  
$$ LANGUAGE plpgsql;
```

Trigger:

```
CREATE TRIGGER trg_student_insert  
AFTER INSERT ON students  
FOR EACH ROW  
EXECUTE FUNCTION log_student_insert();
```

10. Views

```
CREATE VIEW high_scorers AS  
  
SELECT name, marks FROM students WHERE marks > 80;
```