[[Ch04 – CockroachDB SQL]]

**== CockroachDB SQL**

The language of CockroachDB is SQL. While there are some command line utilities, all interaction between an application and the database are mediated by SQL language commands.

SQL is a rich language with a long history – we touched upon some of that history in Chapter 1. A full definition of all SQL language features would require a book in its own right, and would be almost instantly out of date, since the SQL language evolves with each release.

Therefore, this chapter aims to provide you with a broad overview of the SQL language used in CockroachDB without attempting to be a complete reference. We’ll take a task oriented approach to the SQL language, covering the most common SQL language tasks with particular reference to unique features of the CockroachDB SQL implementation.

A complete reference for the CockroachDB SQL language can be found in the [CockroachDB documentation set](https://www.cockroachlabs.com/docs/stable/index.html) . A broader review of the SQL language can be found in the OReilly book “SQL in a Nutshell”. book “SQL in a Nutshell”.

We’ll mainly use the MOVR sample dataset in this chapter to illustrate various SQL language features. To install MOVR….

# SQL Language compatibility

CockroachDB is broadly compatible with the PostgreSQL implementation of the SQL:2003 standard. The SQL:2003 standard contains a number of independent modules and no major database implements all of the standard.

CockroachDB varies from PostgreSQL primarility in the following areas….

* No support for stored procedures, triggers or events
* No support for UDFs
* XML functions
* FULLTExt functions and indexes
* CockroachDB doesn’t differentiate between different sized data types

# Creating tables and indexes

## Creating tables

We’ll discuss the key considerations for database design in Chapter 5. For now, let’s create a few simple tables.

We use CREATE TABLE to create a table within a database. The CREATE TABLE statement must define the columns that occur within the table, and can optionally define indexes, column families , contraints, and partitions associated with the table.

### Column Definitions

#### Datatypes

#### Primary keys

##### Selecting a primary key timestamps, vs uuid

* + Autogenerated primary keys

#### foreign keys

#### Nulls

#### ConstrainTs

## Indexes

Indexes can be created by the CREATE INDEX statement, or an INDEX definition can be included within the CREATE TABLE statement.

We talked a lot about indexes in Chapter 2, and we’ll keep discussing indexes in the schema design and performance tuning chapters.

### Special CockroachDB indexing options

#### inverted indexes

#### STORING clause

#### hash sharded indexes

#### Spatial indexes

### Indexed COLUMN SORT ORDER

## CReate table like

## CREATE TABLE AS SELECT

## ALTERING Tables

## DROPPING tables

# Inserting data

## The Insert statement

### Returning clause

### On Conflict

### DISTINCT ON

## IMPORT/IMPORT INTO

# modifying data

## UPDATE

### FROM clause

### RETURNING

## UPSERT

# deleting data

## DELETE

## TRUNCATE

# Transactions

## BEGIN

## COMMIT/ROLLBACK

## SAVEPOINTS

## Select for Update

## As of system time

# querying data

The SELECT statement is the workhorse of relational query and has a complex and rich syntax. The CockroachDB SELECT statement implements the standard features of the standard SELECT, with just a few CockroachDB-specific features.

## Inline views

## Common Table Expressions

## SELECT list expressions

## FROM clause – index selection

## Joins

## WHERE Clause

## GROUP and HAVING

## ORDERING

## WINDOW FUNCTIONS

# Advanced DDL

In the next chapter, we’ll look at some of the considerations for data modelling and schema design. Some of the following commands are more relevant in that context. However, let’s look at some of the more advanced commands that we can use to define data and indexing.

## Schemas

## Sequences

## Change feeds

## Partitions

## TYPES

# administrative commands

CockroachDB supports commands to maintaine authentication of users and their authorities to perform database operations. It also has a job scheduler that can be used to schedule backup and restore and timed Data definition changes. Other commands support the maintenance the cluster topology

## Cluster maintanence

### Configure zone

### SET CLUSTER SETTING

### SET LOCALITY

### SHOW RANGES

### SPLIT AT

### SURVIVE

## Scheduling and backups

### Backup

### create schedule for backup

### SHOW/CANCEL/PAUSE JOBS

## Security

### create/alter/drop user

### create/alter/drop role

## General Administration

### Show statistics

### SHOW TRACE FOR SESSION

### SHOW TRANSACTIONS

### SHOW/CANCEL SESSION

### EXPLAIN

# Summary