



Querying Data with SQL (DDL)

Session 2

Database for Big Data Analytics

Siti Mariyah, Ph.D.
Politeknik Statistika STIS



Email: sitimariyah@stis.ac.id
Github: <https://github.com/diahnuri>

Session 2 – Querying Data with SQL (DDL)

- Goal: Learn core SQL skills for analysis
- Content:
 - What is SQL?
 - Purpose and Importance of SQL
 - Database Languages
 - SQL Data Definition Language (DDL)

Hands-on:

- Create your first table, insert, and query simple data
- Explore MPD sample data (e.g., call detail records or anonymized logs)
- Write queries to select, count, aggregate, and filter records.

What is SQL?

- SQL is a transform-oriented language, that is, a language designed to use relations to transform inputs into required outputs
- It has 2 major components:
 - A DDL for defining database structure.
 - A DML for retrieving and updating data.
- SQL is relatively easy to learn:
 - It is non-procedural - you specify what information you require, rather than how to get it;
 - It is essentially free-format.

What is SQL?

- An ISO standard now exists for SQL, making it both the formal and de facto standard language for relational databases.
- Consists of standard English words:
 - CREATE TABLE Staff(staffNo VARCHAR(5),
IName VARCHAR(15),
salary DECIMAL(7,2));
 - INSERT INTO Staff VALUES ('SG16', 'Brown', 8300);
 - SELECT staffNo, IName, salary FROM Staff WHERE salary > 10000;
- Can be used by range of users including DBAs, management, application developers, and other types of end users.

Writing SQL Commands

- SQL statement consists of reserved words and user-defined words.
- Reserved words are a fixed part of SQL and must be spelt exactly as required and cannot be split across lines.
- User-defined words are made up by user and represent names of various database objects such as relations, columns, views.
- Most components of an SQL statement are case insensitive, except for literal character data.
 - e.g., search with ‘SMITH’ will not return ‘Smith’
- More readable with indentation and lineation:, e.g.,
 - Each clause should begin on a new line.
 - Start of a clause should line up with start of other clauses.
 - If clause has several parts, should each appear on a separate line and be indented under start of clause.

Literals

- Literals are constants used in SQL statements.
- All non-numeric literals must be enclosed in single quotes (e.g. ‘London’).
- All numeric literals must not be enclosed in quotes (e.g. 650.00).

ISO SQL Data Types

Table 6.1 ISO SQL data types.

| Data type | Declarations | | | | |
|---------------------|------------------------|-------------|---------------------|----------|--|
| boolean | BOOLEAN | | | | |
| character | CHAR | VARCHAR | | | |
| bit | BIT | BIT VARYING | | | |
| exact numeric | NUMERIC | DECIMAL | INTEGER | SMALLINT | |
| approximate numeric | FLOAT | REAL | DOUBLE PRECISION | | |
| datetime | DATE | TIME | TIMESTAMP | | |
| interval | INTERVAL | | | | |
| large objects | CHARACTER LARGE OBJECT | | BINARY LARGE OBJECT | | |

SQL Scalar Data Types

- Boolean Data
 - Consists of two distinct truth values: TRUE and FALSE
- Character Data
 - Can be specified max number of characters (default = 1)
 - Can be either Fixed (e.g., branchNo CHAR(4)) or
 - Varying (e.g., address VARCHAR(30))
- Exact Numeric Number
 - Exact representation of a number: precision and scale
 - DECIMAL[precision [,scale]] (max 99,999,999), e.g., salary DECIMAL(7,2)
 - SMALLINT (max 32,767), e.g., rooms SMALLINT
- Approximate Numeric Data
 - Numbers that do not have exact representation
 - FLOAT [precision]
 - REAL
 - DOUBLE PRECISION
- See here: http://www.w3schools.com/sql/sql_datatypes.asp

Data Definition Language (DDL)

- SQL DDL allows database objects such as schemas, domains, tables, views, and indexes to be created and destroyed.
- Main SQL DDL statements are:

CREATE SCHEMA

DROP SCHEMA

CREATE/ALTER DOMAIN

DROP DOMAIN

CREATE/ALTER TABLE

DROP TABLE

CREATE VIEW

DROP VIEW

- Many DBMSs also provide (not in the standard):

CREATE INDEX

DROP INDEX

Data Definition Language (DDL)

- Relations and other database objects exist in an environment.
- Each environment contains one or more catalogs, and each catalog consists of set of schemas.
- Schema is named collection of related database objects.
- Objects in a schema can be tables, views, domains, assertions, collations, translations, and character sets. All have same owner.

Create Schema

CREATE SCHEMA [Name]

AUTHORIZATION CreatorId]

DROP SCHEMA Name [RESTRICT | CASCADE]

- With RESTRICT (default), schema must be empty or operation fails.
- With CASCADE, operation cascades to drop all objects associated with schema in order defined above. If any of these operations fail, DROP SCHEMA fails.

Example: CREATE SCHEMA SqlTests AUTHORIZATION Smith

Create Table

```
CREATE TABLE TableName (
```

```
{ colName dataType [NOT NULL] [UNIQUE]  
[DEFAULT defaultOption]  
[CHECK searchCondition] [...] }
```

```
[PRIMARY KEY (listOfColumns),]
```

```
{ [UNIQUE (listOfColumns),] [...] }
```

```
{ [FOREIGN KEY (listOfFKColumns)  
REFERENCES ParentTableName [(listOfCKColumns)],  
[ON UPDATE referentialAction]  
[ON DELETE referentialAction ]] [...] }
```

```
{ [CHECK (searchCondition)] [...] }
```

```
)
```

Create Table

- Creates a table with one or more columns of the specified dataType.
- With NOT NULL, system rejects any attempt to insert a null in the column.
- Can specify a DEFAULT value for the column.
- Primary keys should always be specified as NOT NULL.
- FOREIGN KEY clause specifies FK along with the referential action.

Create Table

```
CREATE DOMAIN OwnerNumber AS VARCHAR(5)  
CHECK (VALUE IN (SELECT ownerNo FROM PrivateOwner));
```

```
CREATE DOMAIN StaffNumber AS VARCHAR(5)  
CHECK (VALUE IN (SELECT staffNo FROM Staff));
```

```
CREATE DOMAIN PNumber AS VARCHAR(5);
```

```
CREATE DOMAIN PRRooms AS SMALLINT  
CHECK(VALUE BETWEEN 1 AND 15);
```

```
CREATE DOMAIN PRent AS DECIMAL(6,2)  
CHECK(VALUE BETWEEN 0 AND 9999.99);
```

Create Table

```
CREATE TABLE towers (
    tower_id    INT PRIMARY KEY,
    regency     TEXT NOT NULL,
    latitude    NUMERIC(9,6),
    longitude   NUMERIC(9,6)
);
```

```
CREATE TABLE subscribers (
    subscriber_id INT PRIMARY KEY,
    home_regency  TEXT NOT NULL,
    home_tower_id INT NOT NULL REFERENCES towers(tower_id),
    sim_type      TEXT CHECK (sim_type IN ('prepaid','postpaid'))
);
```

Alter Table

- Add a new column to a table.
- Drop a column from a table.
- Add a new table constraint.
- Drop a table constraint.
- Set a default for a column.
- Drop a default for a column.

Alter Table (Example 1)

- Change towers table by removing default of ‘Denpasar’ for regency column.

ALTER TABLE towers

ALTER regency DROP DEFAULT;

- Change cdr_events table by setting default for event_type column to “data”.

ALTER TABLE cdr_events

ALTER event_type SET DEFAULT “data”;

Alter Table (Example 2)

- Remove constraint from RegencyMustBeInIndonesia, for example if the mobile network operator spans its tower coverage to overseas

ALTER TABLE towers

DROP CONSTRAINT RegencyMustBeInIndonesia;

- Remove constraint from RegencyMustBeInIndonesia, for example if the mobile network operator spans its tower coverage to overseas

ALTER TABLE cdr_events

ADD AllEventTypes event_type;

Drop Table

- `DROP TABLE TableName [RESTRICT | CASCADE]`
 - e.g., `DROP TABLE towers`
- Removes named table and all rows within it.
 - Use `DELETE FROM towers` to delete rows only
- With RESTRICT, if any other objects depend for their existence on continued existence of this table, SQL does not allow request.
- With CASCADE, SQL drops all dependent objects (and objects dependent on these objects).



20

Hands-on

Create Schema

- . Create Schema named MPD

```
CREATE SCHEMA IF NOT EXISTS mpd;
```

Create Table towers

```
DROP TABLE IF EXISTS towers CASCADE;  
CREATE TABLE towers (  
    tower_id    INT PRIMARY KEY,  
    regency     TEXT NOT NULL,  
    latitude    NUMERIC(9,6),  
    longitude   NUMERIC(9,6)  
);
```

Create Table subscribers

```
DROP TABLE IF EXISTS subscribers CASCADE;  
CREATE TABLE subscribers (  
    subscriber_id INT PRIMARY KEY,  
    home_regency TEXT NOT NULL,  
    home_tower_id INT NOT NULL REFERENCES towers(tower_id),  
    sim_type      TEXT CHECK (sim_type IN ('prepaid','postpaid'))  
);
```

Create Table cdr_events

```
DROP TABLE IF EXISTS cdr_events CASCADE;  
CREATE TABLE cdr_events (  
    event_id      BIGINT PRIMARY KEY,  
    subscriber_id INT NOT NULL REFERENCES subscribers(subscriber_id),  
    tower_id      INT NOT NULL REFERENCES towers(tower_id),  
    event_type    TEXT CHECK (event_type IN ('call','sms','data')),  
    event_ts      TIMESTAMP NOT NULL  
);
```

Create Table tourism_events

```
DROP TABLE IF EXISTS tourism_events CASCADE;  
CREATE TABLE tourism_events (  
    event_id    INT PRIMARY KEY,  
    event_name   TEXT NOT NULL,  
    regency     TEXT NOT NULL,  
    start_date  DATE NOT NULL,  
    end_date    DATE NOT NULL  
);
```



Thank You

