

Advanced SQL

Lecture 7c



Outline

- Concepts
 - Transactions: **BEGIN, COMMIT/ROLLBACK**
 - Indexes, [Materialized] Views
 - Assertions, Triggers
- Schema
 - Common data types
 - DDL: **CREATE, ALTER, DROP**
 - Authentication/Authorization
 - **GRANT, REVOKE, WITH GRANT OPTION**



Caution

Note that the specific syntax/functionality of all commands in this lecture are highly dependent upon the chosen DBMS (and possibly even the version).

These slides should be taken as an overview of common options; actual implementation should reference DBMS documentation.



Transactions

- Review: ACID
- By default, each individual query is a transaction
- To group multiple operations:
 - Start: **BEGIN**
 - End: **COMMIT** (default) or **ROLLBACK**



Index

- Supplementary data structure used to make some operations faster
- Defined on a sequence of field(s) of a single table
 - May optionally enforce uniqueness
- More detail in physical tuning
 - When to use, types, tradeoffs



View

A “virtual” table defined via a `SELECT` query over *base table(s)* and/or other views

Common uses

- Convenience/code re-use: if multiple queries rely upon a common data transformation
- Security: users only see the data they “need” to see (e.g. calculation/join/aggregation over base data)
- Performance: a view may optionally be **materialized**, meaning the DBMS actually stores its contents on disk – can reduce query time via caching complex operations/aggregations (more in physical tuning)



Assertion

Declarative constraint that is outside the scope of *implicit/explicit* constraints

Typically cross-table
– Else **CHECK**



Trigger

Part of an *active database* – specifies actions that automatically occur as a result of database events

Typically composed of three components:

- Database update **event(s)**
- Before/after the event(s) occur, the **condition** that determines if the rule action applies
- The **action** to be taken, typically a set of SQL statements



Schema Specification

Create/edit/delete...

- Database
- Table
- Column
- Data type/domain
- Primary/foreign key
- Index
- Constraint
- Trigger, assertion, view
- User, role, privilege

Schema description is stored in the *catalog* (sometimes represented/accessible as tables).



Database

```
CREATE { DATABASE | SCHEMA }  
[IF NOT EXISTS] database_name;
```

```
DROP { DATABASE | SCHEMA }  
[IF EXISTS] database_name;
```

Common to need a **USE database_name** or similar statement to indicate active database context.



Common Data Types

- **BIT**
- **INT** (capacity, length, signed)
- **REAL/DOUBLE/FLOAT** (size, digits)
- **DATE/TIME/DATETIME/TIMESTAMP**
- **CHAR** (length)
- **VARCHAR** (length)
- **TEXT/CLOB**
- **BINARY/BLOB**



Custom Data Types

- **CREATE DOMAIN**
 - Name, base type, constraint(s) via CHECK

- **CREATE TYPE**



Table Overview

```
CREATE TABLE [IF NOT EXISTS] table_name (  
    column_name1 TYPE [OPTIONS],  
    column_name2 TYPE [OPTIONS],  
    {constraint},  
    ...  
);
```

High-level notes

- If an option applies to a single column, it can go with the column; else separate entry, or sometimes separate
- Separate elements may/not have name (for later manipulation)
- Remove table: **DROP TABLE table_name;**
- Alter entry: **ALTER TABLE table_name ADD/ALTER/DROP ...;**



Common Column Modifiers

- **[NOT] NULL**
- **DEFAULT <value>**
- **UNIQUE**
- **PRIMARY KEY**
- **CHECK <expr>**
- **AUTOINCREMENT**
 - DBMS-specific



Keys

Separate line required if multi-column.

Optional: **CONSTRAINT constraint_name**

PRIMARY KEY (c_name1, c_name2, ...)

FOREIGN KEY

(l_c_name1, l_c_name2, ...)

REFERENCES table_name(f_c_name1, ...)

[ON <DELETE/UPDATE> <CASCADE/SET NULL>]



Indexes

```
CREATE [UNIQUE] INDEX index_name  
ON table_name (c_name1, ...)  
[OPTIONS];
```

Notes

- Ordering of columns is **VERY** important
- Options often refer to the type of index being used (e.g. btree, hash, spatial – **VERY** important)



Assertions

```
CREATE ASSERTION assert_name  
CHECK (multi-table expr);
```



Triggers

```
CREATE TRIGGER trigger_name  
<BEFORE/AFTER> <INSERT/UPDATE/DELETE>  
ON table_name FOR EACH ROW  
{body};
```



Views

```
CREATE VIEW view_name  
AS SELECT ...  
[WITH CHECK OPTION];
```



Discretionary Access Control

- Create/remove users
 - **CREATE USER ...**
 - **DROP USER ...**
- Grant/revoke privilege(s)
GRANT/REVOKE <privilege list>
ON <database/table>
TO/FROM user
- **WITH GRANT OPTION** supports propagation of grants

