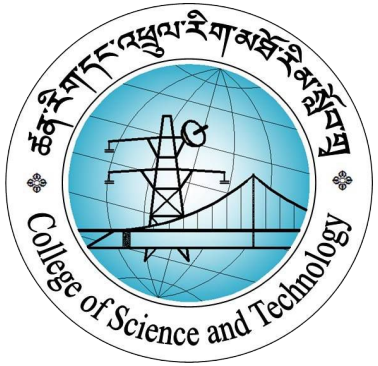


Royal University of Bhutan

DBS101 Database Systems Fundamentals

Lesson 6

Learning Outcomes

1. Implement different types of joins in SQL.
2. Create and implement SQL views.
3. Understand SQL transactions.
4. Implement integrity constraints in SQL to maintain data consistency.
5. Implement indexes using SQL statements.
6. Define SQL data types and schemas.

Join Expressions

Natural Join:

- performs the Cartesian product
- finds consistent tuples and deletes inconsistent tuples
- it deletes the duplicate attributes.
- To perform natural join there must be one common attribute(Column) between two tables.

Join Expressions

Outer Join:

Outer joins are joins that return matched values and unmatched values from either or both tables.

It preserves those tuples that would be lost in a join by creating tuples in the result containing null values.

Join Expressions

Types of outer joins:

- **LEFT JOIN** returns only unmatched rows from the left table, as well as matched rows in both tables.
- **RIGHT JOIN** returns only unmatched rows from the right table , as well as matched rows in both tables.
- **FULL OUTER JOIN** returns unmatched rows from both tables,as well as matched rows in both tables.

Join Expressions

Join Types and Conditions:

Inner Join: the join operations that do not preserve non matched tuples are called inner-join operations

The default join type, when the join clause is used without the outer prefix, is the inner join.

Similarly, natural join is equivalent to natural inner join

View

It is not always desirable for all users to see the entire set of relations in the database.

Then we use the SQL authorization mechanism to restrict access to relations.

View

In SQL, a view is a virtual table based on the result-set of an SQL statement.

View Definition

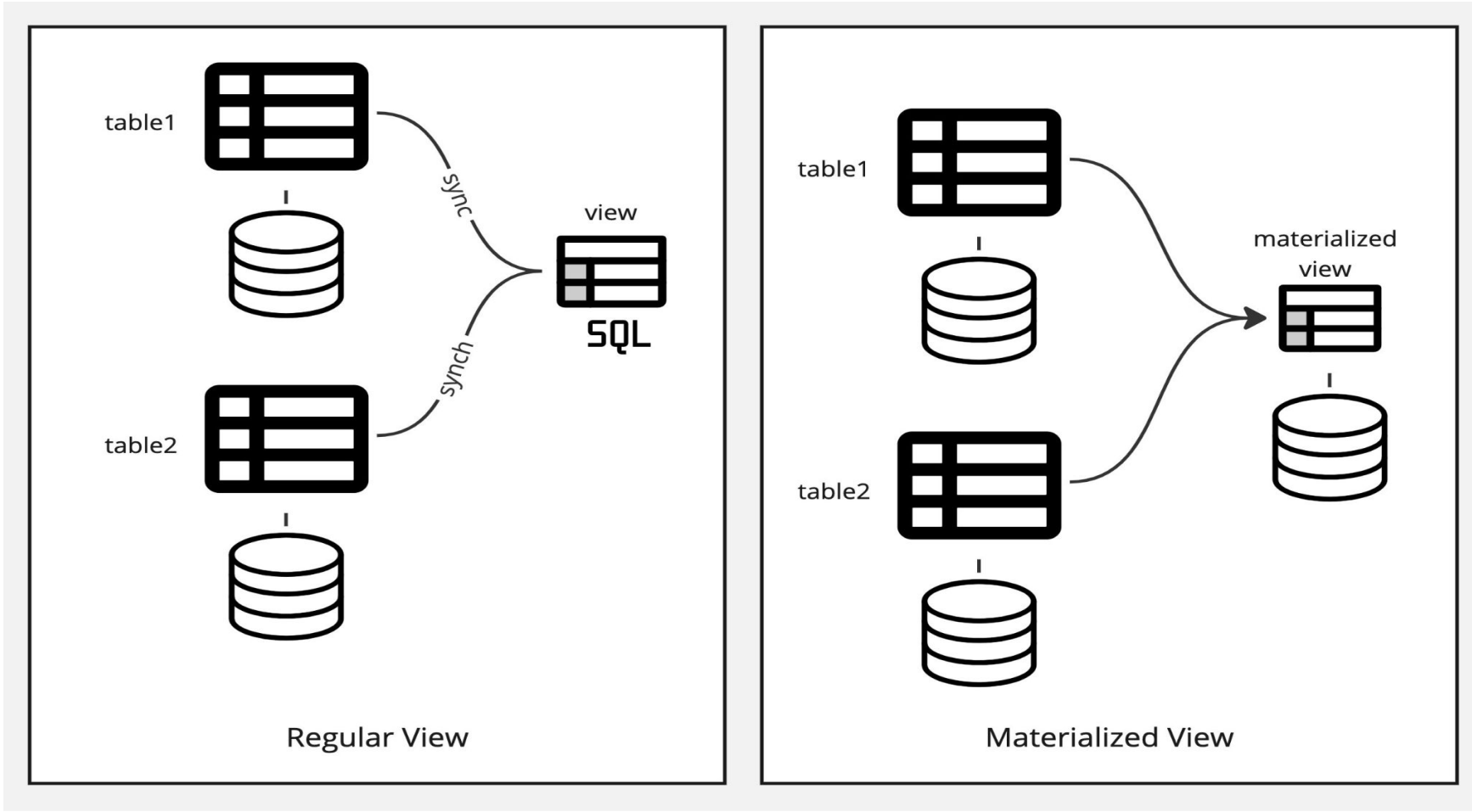
```
create view faculty as
select ID, name, dept name
from instructor;
```

Materialized Views

Certain database systems allow view relations to be stored, but they make sure that, if the actual relations used in the view definition change, the view is kept up-to-date. Such views are called **materialized views.**

The process of keeping the materialized view up-to-date is called **materialized view maintenance.**

Materialized Views



Update of a View

A modification to the database expressed in terms of a view must be translated to a modification to the actual relations in the logical model of the database.

A SQL view is said to be updatable if:

- The from clause has only one database relation.
- The select clause contains only attribute names of the relation and does not have any expressions, aggregates, or distinct specification.
- Any attribute not listed in the select clause can be set to null; that is, it does not have a not null constraint and is not part of a primary key.
- The query does not have a group by or having clause.

Transactions

A transaction consists of a sequence of query and/or update statements.One of the following SQL statements must end the transaction:

Commit work commits the current transaction; it makes the updates performed by the transaction become permanent in the database. After the transaction is committed, a new transaction is automatically started.

Transactions

A transaction consists of a sequence of query and/or update statements.One of the following SQL statements must end the transaction:

Rollback work causes the current transaction to be rolled back;it undoes all the updates performed by the SQL statements in the transaction. Thus, the database state is restored to what it was before the first statement of the transaction was executed.

Integrity Constraints

Integrity constraints ensure that changes made to the database by authorized users do not result in a loss of data consistency

Integrity constraints are usually identified as part of the database schema design process and declared as part of the create table command used to create relations.

SQL Data Types and Schemas

Date and Time Types in SQL

Data Type	Details
date	0001-01-01 to 9999-12-31 to nearest day
time	00:00:00.0000000 to 23:59:59.9999999 with 100 nanosecond accuracy
datetime2	Same as date and time combined
datetimeoffset	As per datetime2 plus timezone offset -14:00 to +14:00
datetime	1753-01-01 to 9999-12-31 to nearest 3.33 milliseconds
smalldatetime	1900-01-01 to 2079-06-06 to 1 minute

SQL Data Types and Schemas

Type Conversion and Formatting Functions:

Although systems perform some data type conversions automatically, others need to be requested explicitly.

```
select cast(ID as numeric(5)) as inst id
from instructor
order by inst id
```

SQL Data Types and Schemas

Default Values:

```
create table student(
   ID varchar (5),
   name varchar (20) not null,
   dept name varchar (20),
   tot cred numeric (3,0) default 0,
   primary key (ID));
```

SQL Data Types and Schemas

Large-Object Types:

To store attributes whose domain consists of large data items such as a photo, a high-resolution medical image, or a video.

large-object data types:

- for character data (clob) and
- binary data (blob).

SQL Data Types and Schemas

```
Large-Object Types:
book review clob(10KB)
image blob(10MB)
movie blob(2GB)
```

User Defined Types

SQL supports two forms of user-defined data types. The first form, which we cover here, is called **distinct types**. The other form, called **structured data types**, allows the creation of complex data types with nested record structures, arrays, and multisets.

The create type clause can be used to define new types. For example, the statement:

```
create type Dollars as numeric(12,2) final;
```

User Defined Types

Domain: A domain is essentially a data type with optional constraints (restrictions on the allowed set of values).

Domains are useful for abstracting common constraints on fields into a single location for maintenance.

User Defined Types

Generating Unique Key Values:

Database systems offer automatic management of unique key-value generation.

Example

ID number(5) generated always as identity

Create Table Extensions

Applications often require the creation of tables that have the same schema as an existing table.

```
create table temp instructor like instructor;
```

Schemas Catalogs and Environments

Hierarchical Organization: Database systems have evolved from flat namespaces to hierarchical structures with catalogs, schemas, and relations.

User Connectivity: Users must connect to a database, providing a username and password, with each user having a unique default catalog and schema.

Three-Part Naming: Relations are identified using a three-part convention: catalog, schema, and relation name, with defaults applied if parts are omitted.

Schemas Catalogs and Environments

Independence and Clashes: Multiple catalogs and schemas enable users and applications to work independently without concerns about naming conflicts.

Schema Management: Schemas can be created and dropped using SQL statements, often automatically created upon user account creation, but catalog management is system-specific and not standardized in SQL.

To be Continued….