

Structured Query Language (Part 1)

Relational Data Structure and Concepts

- Two-dimensional tables whose attributes values are atomic.
- At every row-and-column position within the table, there always exists precisely one value, never a list of values. (Relations do not contain repeating groups after 1NF)
- There are no duplicate rows (There is always a primary key in a relation).
- Rows (tuples) are unordered.

Relational Data Structure and Concepts

- **Degree of Relation** is the number of attributes in relation.
- **Cardinality of Relation** is the number of tuples in relation.
- **Domain** is a pool of values, from which one or more attributes draw their actual values.
 - ◆ For example, a domain of supplier numbers is the set of all possible supplier numbers and the domain.

S#	SNAME	STATUS	CITY
S1	Smith	21	London
S2	Jones	10	Paris
S3	Mary	30	Paris
S4	Clark	20	Athens

The Entity Integrity Rules

- No component of the primary key of a relation is allowed to accept nulls.
- Example
 - ◆ Suppose in the supplier relation, there is a tuple for which the “supplier number” is null.
 - ◆ This would be like saying that there was a supplier in the real world that had no identity.

The Referential Integrity Rule

- The database must not contain any unmatched foreign key values.
- The referential integrity rule says that if B references A, then A must exist.

Consideration in Foreign Key (1)

- Can that foreign key accept nulls?
 - ◆ Example
 - ◆ Does it make sense for a shipment to exist for which the supplier is unknown?

Consideration in Foreign Key (2)

- What should happen on an attempt to delete the target of a foreign key references?
 - ◆ Example
 - ◆ An attempt to delete a supplier for which there exists at least at one matching shipment?

Restricted	The delete operation is "restricted" to the case where there are no such matching shipments (it is rejected otherwise)
Cascades	The delete operation "cascades" to delete those matching shipments also
Nullifies	The foreign key is set to null in all such matching shipments and the supplier is then deleted (of course, this case could not apply if the foreign key cannot accept nulls in the first place)

Consideration in Foreign Key (3)

- What should happen on an attempt to update the primary key of a foreign key reference?
 - ◆ Example
 - ◆ An attempt to update the supplier number for a supplier for which there exists at least one matching shipment?

Restricted	The update operation is "restricted" to the case where there are no such matching shipments (it is rejected otherwise)
Cascades	The update operation "cascades" to update the foreign key in those matching shipments also
Nullifies	The foreign key is set to null in all such matching shipments and the supplier is then updated (of course, this case could not apply if the foreign cannot accept nulls in the first place)

SQL and Relational Model

- SQL (Structured Query Language) is a relational language, which provides an interface between the user and the RDBMS.
- SQL's capabilities are divided between the **Data Description Language (DDL)** and the **Data Manipulation Language (DML)**.

Examples of DDL & DML

DDL	DML
Create tables	Query table data
Modify tables (ALTER)	Modify table data
Create indexes	• Insert
Drop tables	• Delete
Drop indexes	• Update
Authorisation (GRANT)	Aggregate functions
Deauthorisation (REVOKE)	

Relational Data Definition in SQL

Data Types	
DECIMAL (m, n)	Signed numbers, where m is the total number of digits to the right of the decimal point
INTEGER	Large (up to 11 digits) positive or negative whole numbers
SMALL	Small (5 or 6 digits, depending on the DBMS) positive or negative whole numbers. By specifying this data type, less storage space is required
FLOAT (m, n)	Whole and fractional numbers represented in scientific notation, where m is the total number of digits (including sign), and n is the number of digits to the right of the decimal point
CHAR (n)	Alphanumeric (character) data, where n is the maximum length for the character string; n character positions are allocated to each instance of a CHAR column
DATE	Calendar dates. A system variable can be used to set the format for the date data type
LOGICAL	True or false values

Data Description Language (DDL)

- The purpose of the DDL is to create entries in the catalogue (sometimes called the data dictionary or, in some large systems, the repository).
- The catalogue is managed by the RDBMS and it is a requirement of the Relational Model that the catalogue itself takes the form of normalized tables.
- The update of these tables is quite transparent to the user.

Common DDL Command

DDL Commands	Description
CREATE TABLE	Define the structure of a new table
INSERT	Places a new row in a table based on values supplied in the statement, copies one or more rows derived from other database into a table, extracts data from one table and inserts them into another.
UPDATE	Changes values in one or more specified rows of a table by replacing current values with constant or the results of calculation.
DELETE	Removes one or more rows from a table.
CREATE VIEW	Names the view and optionally specifies column names to be used in place of those of the base table.
CREATE INDEX	Defines an index on one column that enables rapid access to the rows of a table in a sequence or randomly by the key value.
ALTER TABLE	Adds columns to an existing.
DROP TABLE	Deletes table.
DROP INDEXES	Deletes index.
GRANT	Granting user access.
REVOKE (REBA privileges)	Removing privilege given to a user.

13

Creating Tables

- Syntax:
 - ◆ CREATE TABLE tablename
(colname1 type(width),
colname2 type(width)
:
colnameN type(width));
- Example:
 - ◆ CREATE TABLE SUPPLIER
(SNO CHAR(2),
SNAME CHAR(5),
STATUS NUMBER(5));

SUPPLIER

DB212 © Peter Lo 2004

SNO | SNAME | STATUS

Altering Tables – Add New Column

- Syntax:
 - ◆ ALTER TABLE tablename
ADD (colname1 type(width),
...
colnameN type(width));
- Example:
 - ◆ ALTER TABLE SUPPLIER
ADD (CITY CHAR(4));

SUPPLIER

DB212 © Peter Lo 2004

SNO | SNAME | STATUS | CITY |

Altering Tables – Modify Column

- Modify data type, size of an existing column
- Syntax:
 - ◆ ALTER TABLE tablename
MODIFY (colname1 type(width),
...
colnameN type(width));
- Example:
 - ◆ ALTER TABLE SUPPLIER
MODIFY (CITY CHAR(6));

DB212 © Peter Lo 2004

16

Deleting Tables

- Once a table is dropped, all the data will be deleted permanently.
- Syntax
 - ◆ DROP TABLE tablename;
- Example:
 - ◆ DROP TABLE SUPPLIER;

Indexing Tables

- Indexing tables helps the DBMS query large tables more quickly and efficiently.
- Syntax
 - ◆ CREATE INDEX indexname ON tablename (colname1, . . . colnameN);
- Data are normally inserted first before creating an index.
- The reverse must be done if your index is unique (each value in a column is unique).
- Example:
 - ◆ CREATE UNIQUE INDEX SUPP_NO ON SUPPLIER (SNO);

Deleting Indexes

- Syntax
 - ◆ DROP INDEX indexname;
- Example:
 - ◆ DROP INDEX SUPP_NO;

Views

- A **View** is also called a **Virtual Table** because it does not actually contain stored data, combines data that is selected from one or more base tables of the other views.
- If the data in the base tables changes, the information displayed using the view also changes.
- Provide simplicity to see exactly what you need; at the same time providing the security to prevent unauthorized users from seeing certain information.
 - ◆ We can use a view to let someone access just the rows and columns that the person needs from base tables.
 - ◆ We can omit columns containing irrelevant or sensitive information.

Creating Views

- Syntax:
 - ◆ CREATE VIEW viewname AS
SELECT colname1, ... colnameN
FROM tablename1, ...tablenameN
WHERE (condition expression);
- Example
 - ◆ CREATE VIEW VENDOR AS
SELECT SNO, SNAME, CITY
FROM SUPPLIER;

Deleting Views

- When we delete a view, the table on which the view is deleted will not be deleted.
- Syntax:
 - ◆ DROP VIEW viewname;
- Example:
 - ◆ DROP VIEW VENDOR

Inserting Records

- Syntax:
 - ◆ INSERT INTO tablename (colname1, ... colnameN)
VALUES (...);
- Example:
 - ◆ INSERT INTO SUPPLIER
VALUES (S2, JONES, 10);
- Value of column would be NULL if it is not included in the INSERT clause.

SUPPLIER			
SNO	SNAME	STATUS	CITY
S1	SMITH	20	LONDON
S2	JONES	10	

Updating Data

- Syntax
 - ◆ UPDATE tablename
SET colname1 = value,
... colnameN = value
WHERE (condition expression);
- Example:
 - ◆ UPDATE SUPPLIER
SET CITY='ROME',
WHERE (SNO = 'S2');

SUPPLIER			
SNO	SNAME	STATUS	CITY
S1	SMITH	20	LONDON
S2	JONES	10	ROME

Update Data (Delete Attribute)

- Deletion of selective attributes (or data) is done by updating the column to NULL.
- Example:
 - ◆ UPDATE SUPPLIER
SET CITY = NULL,
WHERE (SNAME = 'BLAKE');

Delete Record

- Removing a tuple.
- Syntax:
 - ◆ DELETE FROM tablename
WHERE (condition expression);
- Example:
 - ◆ DELETE SUPPLIER
WHERE (SNAME = 'BLAKE');

Controlling Changes

- Use COMMIT command to make changes permanent.
- Example of explicit COMMIT:
 - ◆ UPDATE SUPPLIER
SET CITY = 'PARIS',
WHERE (CITY = 'ROME');
 - ◆ COMMIT;
- Without evoking COMMIT, your record(s) may not be updated.

Controlling Changes

- Some SQL commands cause implicit COMMIT effect.
- SQL commands that cause implicit COMMIT:

ALTER	AUDIT	COMMENT
CONNECT	CREATE	DISCONNECT
DROP	EXIT	NOAUDIT
REVOKE	QUIT	RENAME
- You can perform automatic COMMIT by:
 - ◆ SET AUTOCOMMIT ON
- Note that this cause changes to take immediate effect after INSERT, UPDATE or DELETE is evoked.

Canceling Changes

- Using ROLLBACK command cancels all pending changes
- The database is restored to its state at the time of the last COMMIT.
- Example:
 - ◆ ROLLBACK
- This would not have any effect if you have performed a COMMIT before the ROLLBACK.

Data Security

- Database administrator has all privileges.
- Privileges:
 - ◆ resource – logon, create tables
 - ◆ connect – logon, query only
- Syntax
 - ◆ GRANT<privileges>
TO username
IDENTIFIED BY password;
- Example:
 - ◆ GRANT connect, resource TO SCOTT
IDENTIFIED BY TIGER;

Data Security

- You can permit a user to query and make changes to your tables.
- Example:
 - ◆ GRANT SELECT, INSERT
ON SUPPLIER TO SCOTT;
- Table owner can grant them privileges:

SELECT	INSERT	UPDATE
DELETE	ALTER	INDEX
- To grant a user all privileges:
 - ◆ GRANT ALL ON SUPPLIER TO SCOTT;

Data Security

- You can give other user to grant privileges for one of your tables.
- Example:
 - ◆ GRANT ALL ON SUPPLIER
TO SCOTT WITH GRANT OPTION;
- Revoking for system privileges can only be done by the database administrator.
- Table privileges can be revoked any time by the respective users.
- Example:
 - ◆ REVOKE INSERT ON SUPPLIER FROM SCOTT;

Query Database

- Syntax:
 - ◆ SELECT colname1, ..., colnameN
FROM tablename1, ..., tablenameN
WHERE (condition expression)
GROUP BY (condition expression)
HAVING (condition expression)
ORDER BY colname1, ..., colnameN;

Query Database - Projection

- Selecting all columns:
 - ◆ SELECT * FROM SUPPLIER;
- Selecting specific columns:
 - ◆ SELECT SNO, SNAME, CITY FROM SUPPLIER;
- This is also known as projection.

Query Database - Restriction

- Selecting specific records or tuples (Restriction)
 - ◆ SELECT * FROM SUPPLIER
WHERE CITY = 'LONDON';
- Selecting specific atomic data:
 - ◆ SELECT SNO, SNAME, CITY FROM SUPPLIER
WHERE CITY = 'LONDON';
- Selecting data BETWEEN a range:
 - ◆ SELECT SNO, PNO, JNO FROM SPJ
WHERE QTY BETWEEN 400 AND 700;

Query Database - IN

- Equal to any member IN the list:
 - ◆ SELECT *
FROM PART
WHERE PNAME IN ('NUT', 'SCREW');
- Another example:
 - ◆ SELECT PNO, PNAME, WEIGHT, CITY
FROM SUPPLIER
WHERE CITY NOT IN ('ROME', 'PARIS');

Query Database – LIKE

- Matching a pattern LIKE %;
 - ◆ SELECT * FROM SUPPLIER
WHERE SNAME LIKE 'SMITH%';
- Supplier names like 'SMITH & SMITH', and 'SMITHEEN' would all be acceptable.
- Example:
 - ◆ SELECT * FROM SUPPLIER
WHERE SNAME LIKE %SMITH%';
- Supplier names like 'ADAM SMITH', 'GOLDSMITH', and 'SMITH & ALICE' would all be acceptable.