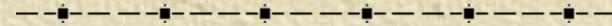


Structured Query Language (SQL)



Prepared by: Mr. Heng Jooi Huang

Structured Query Languages (SQL)

Instructional Objectives:

- ❖ The basic commands and functions of SQL.
- ❖ How to use SQL for data administration (to create tables, indexes, and views).
- ❖ How to use SQL for data manipulation (to add, modify, delete, and retrieve data).
- ❖ How to use SQL to query a database to extract useful information.

1) Introduction To SQL

Ideal Database Language:

- ❖ Create database and table structures.
- ❖ Perform basic data management chores (add, delete, and modify).
- ❖ Perform complex queries to transform data into useful information.
- ❖ Perform basic functions with minimal user effort.
- ❖ Command structure and syntax must be easy to learn.
- ❖ It must be portable.

1) Introduction To SQL

- ❖ SQL functions fit into two broad categories:

1) Data definition language

- ❖ Includes commands to **create the database objects** such as tables, indexes, and views, as well as commands to **define access rights** to those database objects. (CREATE, DROP, ALTER)

2) Data manipulation language

- ❖ Includes commands to **insert, update, delete, and retrieve data** within database tables. (SELECT, INSERT, UPDATE, DELETE, COMMIT, ROLLBACK)

SQL Data Definition Commands

COMMAND OR OPTION	DESCRIPTION
CREATE SCHEMA	Creates a database schema
AUTHORIZATION	
CREATE TABLE	Creates a new table in the user's database schema
NOT NULL	Ensures that a column will not have null values
UNIQUE	Ensures that a column will not have duplicate values
PRIMARY KEY	Defines a primary key for a table
FOREIGN KEY	Defines a foreign key for a table
DEFAULT	Defines a default value for a column (when no value is given)
CHECK	Constraint used to validate data in an attribute
CREATE INDEX	Creates an index for a table
CREATE VIEW	Creates a dynamic subset of rows/columns from one or more tables
ALTER TABLE	Modifies a table's definition (adds, modifies, or deletes attributes or constraints)
CREATE TABLE AS	Creates a new table based on a query in the user's database schema
DROP TABLE	Permanently deletes a table (and thus its data)
DROP INDEX	Permanently deletes an index
DROP VIEW	Permanently deletes a view

Some Common SQL Data Types

DATA TYPE	FORMAT	COMMENTS
Numeric	NUMBER(L,D)	The declaration NUMBER(7,2) indicates numbers that will be stored with two decimal places and may be up to six digits long, including the sign and the decimal place. Examples: 12.32, -134.99.
	INTEGER	May be abbreviated as INT. Integers are (whole) counting numbers, so they cannot be used if you want to store numbers that require decimal places.
	SMALLINT	Like INTEGER, but limited to integer values up to six digits. If your integer values are relatively small, use SMALLINT instead of INT.
	DECIMAL(L,D)	Like the NUMBER specification, but the storage length is a <i>minimum</i> specification. That is, greater lengths are acceptable, but smaller ones are not. DECIMAL(9,2), DECIMAL(9), and DECIMAL are all acceptable.
Character	CHAR(L)	Fixed-length character data for up to 255 characters. If you store strings that are not as long as the CHAR parameter value, the remaining spaces are left unused. Therefore, if you specify CHAR(25), strings such as "Smith" and "Katzenjammer" are each stored as 25 characters. However, a U.S. area code is always three digits long, so CHAR(3) would be appropriate if you wanted to store such codes.
	VARCHAR(L) or VARCHAR2(L)	Variable-length character data. The designation VARCHAR2(25) will let you store characters up to 25 characters long. However, VARCHAR will not leave unused spaces. Oracle users may use VARCHAR2 as well as VARCHAR.
Date	DATE	Stores dates in the Julian date format. (YYYY-MM-DD)

SQL Data Manipulation Commands - Part 1

COMMAND OR OPTION	DESCRIPTION
SELECT	Selects attributes from rows in one or more tables or views
FROM	Selects one or more tables from database
WHERE	Restricts the selection of rows based on a conditional expression
GROUP BY	Groups the selected rows based on one or more attributes
HAVING	Restricts the selection of grouped rows based on a condition
ORDER BY	Orders the selected rows based on one or more attributes
INSERT	Inserts row(s) into a table
UPDATE	Modifies an attribute's values in one or more table's rows
DELETE	Deletes one or more rows from a table
COMMIT	Permanently saves data changes
ROLLBACK	Restores data to their original values

SQL Data Manipulation Commands - Part 2

COMMAND OR OPTION	DESCRIPTION
COMPARISON OPERATORS	
=, <, >, <=, >=, <>	Used in conditional expressions
LOGICAL OPERATORS	
AND/OR/NOT	Used in conditional expressions
SPECIAL OPERATORS	
BETWEEN	Checks whether an attribute value is within a range
IS NULL	Checks whether an attribute value is null
LIKE	Checks whether an attribute value matches a given string pattern
IN	Checks whether an attribute value matches any value within a value list
EXISTS	Checks whether a subquery returns any rows (True-1 or False-0)
DISTINCT	Limits values to unique values
AGGREGATE FUNCTIONS	
COUNT	Returns the number of rows with non-null values for a given column
MIN	Returns the minimum attribute value found in a given column
MAX	Returns the maximum attribute value found in a given column
SUM	Returns the sum of all values for a given column
AVG	Returns the average of all values for a given column

1) Introduction To SQL

- 🐼 SQL is relatively easy to learn.
- 🐼 Basic command set has a vocabulary of less than 100 words.
- 🐼 Nonprocedural language (concentrate on **WHAT** but not **HOW** to do it).
- 🐼 American National Standards Institute (ANSI) prescribes a standard SQL - forms basis for present and future DBMS integration efforts, allowing us to link hierarchical, network, relational and object-oriented databases.

1) Introduction To SQL

- SQL statement consists of **reserved words** (upper case letters) and **user-defined words** (lower case letters or title case letters).
- Reserved words** are a fixed part of the **SQL** language and have a **fixed meaning**. They must be spelt **exactly** as required.

- Example:

Follow created table(s) and attribute(s) name

```
SELECT  
FROM  
GROUP BY  
Bno,  
Staff  
Bno;  
COUNT  
(Sno)  
AS  
Staff_Count
```

The diagram illustrates the decomposition of an SQL query into its components. It shows a sequence of tokens: 'SELECT', 'FROM', 'GROUP BY', 'Bno,' (with a comma), 'Staff' (with a new line), 'Bno;', 'COUNT', '(', 'Sno', ')', 'AS', and 'Staff_Count'. Arrows point from the first three tokens ('SELECT', 'FROM', 'GROUP BY') to a red box labeled 'Reserved Words'. Arrows point from the remaining tokens ('Bno,', 'Staff', 'Bno;', 'COUNT', '(', ')', 'AS', 'Staff_Count') to a blue box labeled 'User-defined Words'. A purple box labeled 'Follow created table(s) and attribute(s) name' spans the entire row of tokens.

Reserved Words

User-defined Words

1) Introduction To SQL

- ❖ Most components of a **SQL statement** are **case insensitive**, but **literal character data** must be typed **exactly** as it appears in the database.
- ❖ Example: if we store a person's name as '**BONNIE**' and then search for it using the string '**Bonnie**', the record will **not be found**.
- ❖ Many dialects of **SQL** require the use of a **statement terminator (semicolon ;)** to mark the end of each **SQL** statement.

2) Data Manipulation Commands – SELECT Statement

- ❖ The purpose of the **SELECT** statement is to **retrieve and display data from one or more database tables**.
- ❖ The **general form** of the **SELECT** statement is:

```
SELECT      [DISTINCT | ALL] {* | [ all column_name ] [ AS new_name ] } [,...]
FROM        table_name [alias] [...]
[ WHERE     condition ] – filters the rows subject to some condition
[ GROUP BY column_list ] – forms groups of rows with the same column value
[ HAVING    condition ] – filters the groups, works together with GROUP BY
[ ORDER BY column_list ] – to arrange the result table' records (output)
```

- ❖ A vertical bar | indicates a choice among alternative.
- ❖ Curly braces { } indicate a required element.
- ❖ Square brackets [] indicate an optional element.
- ❖ An ellipsis ... is used to indicate optional repetition of an item zero or more times.

Example Of Database – DreamHome Rental

- ❖ Example of database used to illustrate the SQL queries statement:

Branch (Bno, Street, Area, City, Pcode, Tel_No, Fax_No)

Staff (Sno, FName, LName, Address, Tel_No, Position, Sex, DOB, Salary, NIN, Bno)

Property_For_Rent (Pno, Street, Area, City, Pcode, Type, Rooms, Rent, Ono, Sno, Bno)

Renter (Rno, FName, LName, Address, Tel_No, Pref_Type, Max_Rent, Bno)

Owner (Ono, FName, LName, Address, Tel_No)

Viewing (Rno, Pno, VDate, VComment)

Instance of the DreamHome rental database.
Branch

Bno	Street	Area	City	Pcode	Tel_No	Fax_No
B5	22 Deer Rd	Sidcup	London	SW1 4EH	0171-886-1212	0171-886-1214
B7	16 Argyll St	Dyce	Aberdeen	AB2 3SU	01224-67125	01224-67111
B3	163 Main St	Partick	Glasgow	G11 9QX	0141-339-2178	0141-339-4439
B4	32 Manse Rd	Leigh	Bristol	BS99 1NZ	0117-916-1170	0117-776-1114
B2	56 Clover Dr		London	NW10 6EU	0181-963-1030	0181-453-7992

National Insurance Number

Staff

Sno	FName	LName	Address	Tel_No	Position	Sex	DOB	Salary	NIN	Bno
SL21	John	White	19 Taylor St, Cranford, London	0171-884-5112	Manager	M	01/10/1980	30000	WK442011B	B5
SG37	Ann	Beech	81 George St, Glasgow PA1 2JR	0141-848-3345	Snr Asst	F	10/11/1960	12000	WL432514C	B3
SG14	David	Ford	63 Ashby St, Partick, Glasgow G11	0141-339-2177	Deputy	M	24/03/1970	18000	WL220658D	B3
SA9	Mary	Howe	2 Elm Pl, Aberdeen AB2 3SU		Assistant	F	19/02/1992	9000	WM532187D	B7
SG5	Susan	Brand	5 Gt Western Rd, Glasgow G12	0141-334-2001	Manager	F	03/06/1977	24000	WK588932E	B3
SL41	Julie	Lee	28 Malvern St, Kilburn NW2	0181-554-3541	Assistant	F	13/06/1995	9000	WA290573K	B5

Yearly Salary

Property_For_Rent

Pno	Street	Area	City	Pcode	Type	Rooms	Rent	Ono	Sno	Bno
PA14	16 Holhead	Dee	Aberdeen	AB7 5SU	House	6	650	CO46	SA9	B7
PL94	6 Argyll St	Kilburn	London	NW2	Flat	4	400	CO87	SL41	B5
PG4	6 Lawrence St	Partick	Glasgow	G11 9QX	Flat	3	350	CO40	SG14	B3
PG36	2 Manor Rd		Glasgow	G32 4QX	Flat	3	375	CO93	SG37	B3
PG21	18 Dale Rd	Hyndland	Glasgow	G12	House	5	600	CO87	SG37	B3
PG16	5 Novar Dr	Hyndland	Glasgow	G12 9AX	Flat	4	450	CO93	SG14	B3

Renter

Rno	FName	LName	Address	Tel_No	Pref_Type	Max_Rent	Bno
CR76	John	Kay	56 High St, Putney, London SW1 4EH	0171-774-5632	Flat	425	B5
CR56	Aline	Stewart	64 Fern Dr, Pollock, Glasgow G42 0BL	0141-848-1825	Flat	350	B3
CR74	Mike	Ritchie	18 Tain St, Gourock PA1G 1YQ	01475-392178	House	750	B3
CR62	Mary	Tregear	5 Tarbot Rd, Kildary, Aberdeen AB9 3ST	01224-196720	Flat	600	B7

Monthly Rental

Owner

Ono	FName	LName	Address	Tel_No
CO46	Joe	Keogh	2 Fergus Dr, Banchory, Aberdeen AB2 7SX	01224-861212
CO87	Carol	Farrel	6 Achray St, Glasgow G32 9DX & 9DY	0141-357-7419
CO40	Tina	Murphy	63 Well St, Shawlands, Glasgow G42	0141-943-1728
CO93	Tony	Shaw	1/2 Park Pl, Hillhead, Glasgow G4 0QR	0141-225-7025

Viewing

Rno	Pno	VDate	VComment
CR56	PA14	24-May-98	too small
CR76	PG4	20-Apr-98	too remote
CR56	PG4	26-May-98	
CR62	PA14	14-May-98	no dining room
CR56	PG36	28-Apr-98	

Sorting Results (ORDER BY Clause)

- ✿ ASC (ascending order) being the default setting.
 - ❖ Example: A, B, C, D, E...or 1, 2, 3, 4, 5...
- ✿ DESC (descending order) must specify whenever is being used.
 - ❖ Example: ...E, D, C, B, A or ...5, 4, 3, 2, 1
- ✿ The ORDER BY clause must always be the last clause of the SELECT statement.
- ✿ It is possible to include more than one element in the ORDER BY clause.
- ✿ The first element in the ORDER BY clause is a major sort key and the subsequent elements in the ORDER BY clause are minor sort keys.

Joining Database Tables

WHY Join Tables?

- Want to retrieve data from > 1 tables at a time.
- Want to check condition(s) from other table(s).
- Want to become bridge between two tables.

HOW?

- Enumerate (name each) table one by one in the FROM clause of the SELECT statement.

Example:

```
SELECT... FROM Staff, Branch, Property_For_Rent  
SELECT... FROM Staff S, Branch B, Property_For_Rent P
```

NOTE:

- Select on rows where there are common attributes. (FOREIGN KEY matches PRIMARY KEY)
- If same attribute appear in more than one table from the JOINED tables, the source table of the attribute must be defined.

Comparison / Mathematical Operators

Equal To

=

Less Than

<

Greater Than

>

Less Than and Equal To

< =

Greater Than and Equal To

> =

Not Equal To

< >

Some SQL version use !=

Logical Operators (OR, AND, NOT)

- ❖ An expression is evaluated left to right.
- ❖ Sub-expressions in bracket are evaluated first.
- ❖ NOTs are evaluated before ANDs and ORs.
- ❖ ANDs are evaluated before ORs.
- ❖ Rules of Precedence: () → NOT → AND → OR

What happen if without bracket ?

```
SELECT Sno, FName, LName, Position, Salary  
FROM Staff  
WHERE Position = 'Manager' OR Position = 'Assistant'  
      AND Sex = 'F';
```



Special Operators

DISTINCT

← Eliminate Duplication

BETWEEN / NOT BETWEEN

← Range Checking

IS NULL / IS NOT NULL

← Empty Checking

LIKE / NOT LIKE

← Pattern Matching

IN / NOT IN

← Return Values

EXISTS / NOT EXISTS

← Return T (1) or F (0)

Special Operators (LIKE / NOT LIKE)

- SQL allows you to use the **percent sign (%)** and **underscore (_)** wildcard characters to make matches when the entire string is not known.

1) % means any and all following characters are eligible.

Examples: '**J%**' includes Johnson, Jones, Jernigan, July, J-231Q; '**Jo%**' includes Johnson and Jones.

2) _ means any one character may be substituted for the underscore.

Examples: '**_o_es**' includes Jones, Cokes, Cones, Roles; '**01_-7552525**' includes 012-7552525, 016-7552525, 019-7552525.

NOTE: MS Access uses “ * ” as “ % ” and uses “ ? ” as “ _ ”.



Aggregate Functions

COUNT() - count

SUM() - sum

MAX() - maximum

MIN() - minimum

AVG() - average

STDDEV() – standard deviation

VARIANCE() – variance

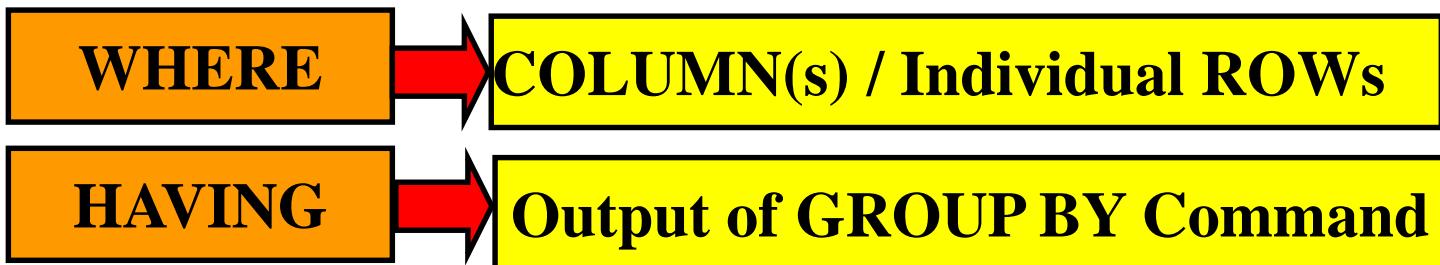
SQL will perform various mathematical summaries, such as **counting the number of rows** that contain a specified condition, **finding the minimum or maximum values** for some specified attribute, **summing the values in a specified column**, and **averaging the values** in a specified column.

Aggregate Functions

- ❖ Aggregate functions operate on a **single column** of the table and return a **single value**.
- ❖ **COUNT**, **MIN** and **MAX** apply to **both numeric and non-numeric fields**, but **SUM** and **AVG** used on **numeric fields only**.
- ❖ **COUNT(*)** is used to count **all the rows** of a table, regardless of whether **nulls** or **duplicate values** occur.
- ❖ Apart from **COUNT(*)**, each function **eliminates nulls first** and **operates only** on the remaining **non-null values**.
- ❖ Aggregate functions can be used **only** in the **SELECT list** and in the **HAVING clause**. WHERE clause 

Restricting GROUP BY Using HAVING

- ❖ HAVING operates like a WHERE.



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK

Data Entry: INSERT Command

- SQL uses the **INSERT command** to enter data into a table.

(i) Single Row Insertion

```
INSERT INTO TableName [ ( ColumnList ) ]  
VALUES ( DataValueList )
```



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK

(ii) Multiple Rows Insertion

INSERT INTO TableName [(ColumnList)]

SELECT... (Can be any valid SELECT statement.)



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK

Making a Correction or Modification: UPDATE Command

- SQL uses the **UPDATE command** to correct or modify data inside the table.
- The UPDATE command's syntax:

UPDATE TableName

SET ColumnName1 = DataValue1 [, ColumnName2 = DataValue2 ...]

[WHERE SearchCondition]



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK

Deleting Table Row(s): DELETE Command

- SQL uses the **DELETE command** to delete row(s) in a table.
- The DELETE command's syntax:

```
DELETE FROM    TableName  
[WHERE    SearchCondition]
```



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK

Saving the Table Contents: COMMIT Command

- ❖ Any changes made to the table contents are not physically saved on disk until we use the COMMIT command.
- ❖ If a power outage or some other interruption occurs before we COMMIT the table, the original table contents are retained.
- ❖ COMMIT often used to save additions, changes and deletions made in the table contents.

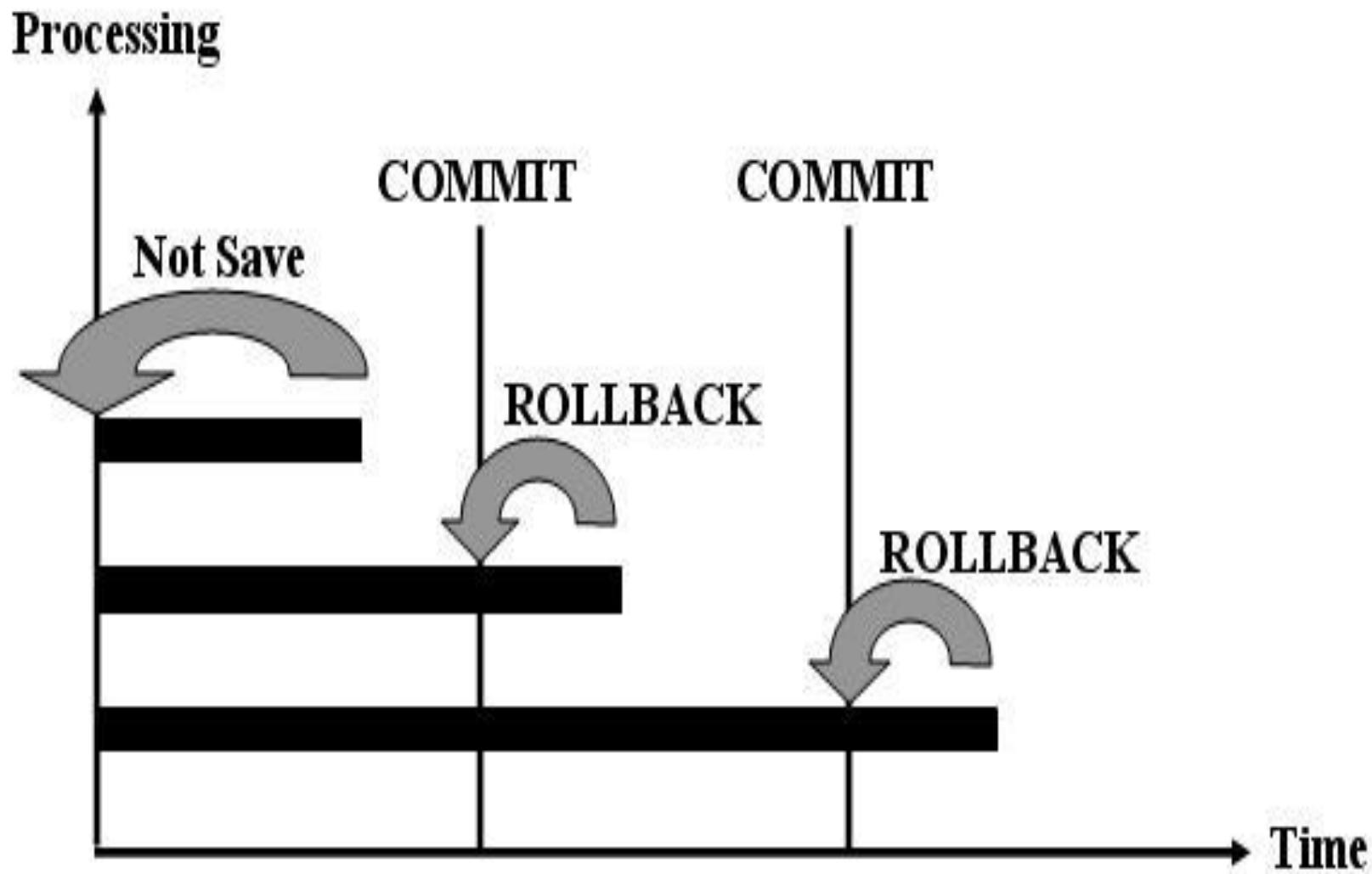
Save the change of table(s).

COMMIT;

NOTE: MS Access does not support the COMMIT command. MS Access automatically saves changes after the execution of each SQL command.



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK



3) Data Manipulation Commands – INSERT, UPDATE, DELETE, COMMIT, ROLLBACK

Restoring the Table Contents: ROLLBACK Command

- ❖ If we have not yet used the COMMIT command to store permanent changes in the table contents, we may restore the database to its previous condition (consistent state) with the ROLLBACK command.
- ❖ ROLLBACK does not require us to specify the table name.
- ❖ SQL assumes the database currently in memory is the one to be restored.

Restore the table.

ROLLBACK; (and press ENTER)



4) Data Definition Commands

– CREATE SCHEMA, DROP SCHEMA

- ❖ The CREATE SCHEMA command's syntax:

```
CREATE SCHEMA [Name | AUTHORIZATION CreatorIdentifier]
```

CREATE SCHEMA DreamHome AUTHORIZATION Eric; /
CREATE SCHEMA AUTHORIZATION Eric;

- ❖ The DROP SCHEMA command's syntax:

```
DROP SCHEMA Name [RESTRICT (default) | CASCADE]
```

- ❖ If **RESTRICT** is specified, the schema **must be empty** or the **operation fails**.

DROP SCHEMA DreamHome; /
DROP SCHEMA DreamHome RESTRICT;

- ❖ If **CASCADE** is specified, the **operation cascades to drop all objects** associated with the schema.

DROP SCHEMA DreamHome CASCADE;

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

- ❖ The CREATE TABLE command's syntax:

CREATE TABLE TableName

(**ColumnName DataType [NULL | NOT NULL] [UNIQUE] [DEFAULT DefaultOption] [CHECK (SearchCondition)] [, ...],**

⋮

[**PRIMARY KEY(PK ColumnName [, ...]),]**

{ **UNIQUE(PK ColumnName [, ...]), }**

{ [**FOREIGN KEY(FK ColumnName) REFERENCES ParentTableName]**
[**ON DELETE ReferentialAction]**
[**ON UPDATE ReferentialAction] });**

- ❖ **Required Data:** **NULL** (default), **NOT NULL**

➤ When **NOT NULL** is specified, the system rejects any attempt to insert a null in the column.

➤ If **NULL** (default) is specified, the system accepts nulls.

➤ Example: **Street** **VARCHAR(15)** **NOT NULL,**
Area **VARCHAR(10),**

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

❖ Domain Constraint: CHECK

- **CHECK** clause that allows a **constraint** to be defined on a **column** or the **entire table**.
- Example without default value:

Sex CHAR NOT NULL CHECK(Sex IN ('M', 'F')),

- Example with default value:

Sex CHAR NOT NULL **DEFAULT 'M'
CHECK(Sex IN ('M', 'F')),**

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

❖ Entity Integrity: PRIMARY KEY

- The **primary key** of a table must contain a **unique, non-null value** for each row.
- The **PRIMARY KEY** clause can be specified **only once** per table.
- Example:

Property_For_Rent Table

Pno CHAR(4) NOT NULL

PRIMARY KEY(Pno),

UNIQUE,

*Not Required

Viewing Table

Rno CHAR(4) NOT NULL,

Pno CHAR(4) NOT NULL,

PRIMARY KEY(Rno, Pno), / UNIQUE(Rno, Pno),

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

❖ Referential Integrity: FOREIGN KEY

- If the **foreign key** contains a **value**, that **value** must **refer** to an existing **primary key** in the **parent table**.
- Example without referential action: define the foreign key Bno of the Staff Table.

FOREIGN KEY(Bno) REFERENCES Branch,

- Example with referential action: define the foreign key Bno of the Staff Table.

**FOREIGN KEY(Bno) REFERENCES Branch
ON DELETE SET NULL ON UPDATE CASCADE,**

- SQL rejects any **INSERT** or **UPDATE** operation that attempts to create a **foreign key value** in a **child table** **without** a matching **primary key value** in the **parent table**.

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

❖ Referential Action: ON DELETE, ON UPDATE

- Any **DELETE** or **UPDATE** operation that attempts to delete or update a foreign key value in the parent table that has some matching rows in the child table is dependent on the referential action specified the ON UPDATE and ON DELETE subclauses of the FOREIGN KEY clause.
- **ON DELETE CASCADE** → Delete the row from the parent table and automatically delete the matching rows in the child table.
- **ON DELETE SET NULL** → Delete the row from the parent table and set the foreign key value(s) in the child table to NULL. This is valid only if the foreign key columns do not have the NOT NULL qualifier specified.

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

- **ON DELETE SET DEFAULT** → Delete the row from the parent table and set each component of the foreign key in the child table to the specified default value. This is valid only if the foreign key columns have a DEFAULT value specified.
- **ON DELETE NO ACTION** → Reject the delete operation from the parent table. This is the default setting if the ON DELETE rule is omitted.
- **ON UPDATE CASCADE** → Foreign key value(s) in the child table are set to the new value(s) of the primary key in the parent table.

3.4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

- ❖ To illustrate the table creation process, we create the structures for the following tables:

(1) Branch Table

CREATE TABLE Branch

Bno	CHAR(2)	NOT NULL,
Street	VARCHAR(15)	NOT NULL,
Area	VARCHAR(10),	
City	VARCHAR(10)	NOT NULL,
Pcode	VARCHAR(10)	NOT NULL,
Tel_No	VARCHAR(13)	UNIQUE,
Fax_No	VARCHAR(13)	UNIQUE,
PRIMARY KEY(Bno);		

(2) Staff Table

CREATE TABLE Staff

(Sno	CHAR(4)	NOT NULL,
FName	VARCHAR(15)	NOT NULL,
LName	VARCHAR(15)	NOT NULL,
Address	VARCHAR(40),	
Tel_No	VARCHAR(13),	
Position	VARCHAR(10)	NOT NULL,
Sex	CHAR	DEFAULT 'M'
	NOT NULL	CHECK(Sex IN ('F', 'M')),
DOB	DATE,	
Salary	NUMBER(5,0)	NOT NULL,
NIN	CHAR(9),	
Bno	CHAR(2),	
	PRIMARY KEY(Sno),	
	FOREIGN KEY(Bno) REFERENCES Branch;	

(3) Property_For_Rent Table

```
CREATE TABLE Property_For_Rent
(Pno CHAR(4) NOT NULL,
Street VARCHAR(15) NOT NULL,
Area VARCHAR(10),
City VARCHAR(10) NOT NULL,
Pcode VARCHAR(10) NOT NULL,
Type VARCHAR(5) DEFAULT 'Flat' NOT NULL,
Rooms SMALLINT DEFAULT 3 NOT NULL
    CHECK(Rooms BETWEEN 1 AND 10),
Rent NUMBER(4,0) DEFAULT 350 NOT NULL
    CHECK(Rent BETWEEN 350 AND 1000),
Ono CHAR(4) NOT NULL REFERENCES Owner(Ono),
Sno CHAR(4),
Bno CHAR(2) NOT NULL,
PRIMARY KEY(Pno),
FOREIGN KEY(Sno) REFERENCES Staff(Sno),
FOREIGN KEY(Bno) REFERENCES Branch,
CONSTRAINT Property_Type CHECK(Type IN ('Flat', 'House')) );
```

(4) Renter Table

CREATE TABLE Renter

Rno	CHAR(4)	NOT NULL,
FName	VARCHAR(15)	NOT NULL,
LName	VARCHAR(15)	NOT NULL,
Address	VARCHAR(40)	NOT NULL,
Tel_No	VARCHAR(13),	
Pref_Type	VARCHAR(5)	DEFAULT 'House' NOT NULL
		CHECK(Pref_Type IN ('Flat', 'House')),
Max_Rent	NUMBER(3,0)	NOT NULL,
Bno	CHAR(2),	
		PRIMARY KEY(Rno),
		FOREIGN KEY(Bno) REFERENCES Branch);

(5) Owner Table

CREATE TABLE Owner

(Ono	CHAR(4)	NOT NULL,
FName	VARCHAR(15)	NOT NULL,
LName	VARCHAR(15)	NOT NULL,
Address	VARCHAR(40)	NOT NULL,
Tel_No	VARCHAR(13),	
	PRIMARY KEY(Ono));	

(6) Viewing Table

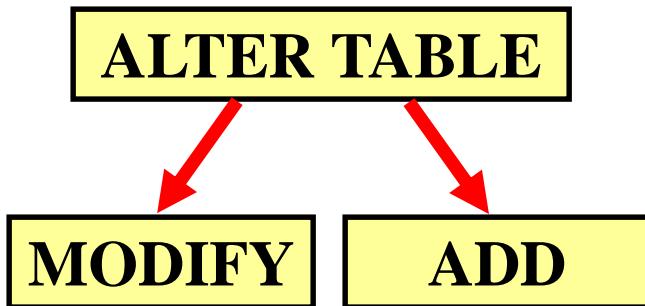
CREATE TABLE Viewing

(Rno	CHAR(4)	NOT NULL,
Pno	CHAR(4)	NOT NULL,
VDate	DATE	NOT NULL,
	VComment VARCHAR(40),	
	PRIMARY KEY(Rno, Pno));	

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

HOW to change table's design view / its structure?



- ❖ ALTER TABLE command with 2 options: MODIFY and ADD.
- ❖ MODIFY – changes column characteristics.

ALTER TABLE <table name>

MODIFY (<column name> <new column characteristic>);

- ❖ ADD – allows adding a column.

ALTER TABLE <table name>

ADD (<new column name> <new column characteristic>);

4) Data Definition Commands

– CREATE TABLE, ALTER TABLE, DROP TABLE

Removing the Table

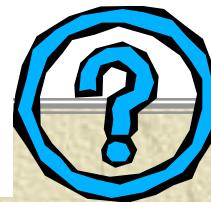
- ❖ Use DROP TABLE command to remove a table from database.
- ❖ The DROP TABLE command's syntax:
DROP TABLE TableName [RESTRICT (default) | CASCADE]
- ❖ If **RESTRICT (default)** is specified and there are **any other objects** that **depend for their existence**, SQL does **not allow** the **DROP TABLE** request to proceed.
- ❖ If **CASCADE** is specified, SQL **automatically drops all dependent objects**.

Delete the Branch table:

DROP TABLE Branch;

OR

DROP TABLE Branch CASCADE;



SQL Statements

SELECT
INSERT
UPDATE
DELETE
MERGE

Data manipulation language (DML)

CREATE
ALTER
DROP
RENAME
TRUNCATE
COMMENT

Data definition language (DDL)

GRANT
REVOKE

Data control language (DCL)

COMMIT
ROLLBACK
SAVEPOINT

Transaction control language (TCL)

Summary

DML

2 Categories of SQL Commands

DDL

COUNT(), SUM(), MAX(), MIN(), AVG()

SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY

DISTINCT
LIKE / NOT LIKE
BETWEEN / NOT BETWEEN
IS NULL / IS NOT NULL
EXISTS / NOT EXISTS
IN / NOT IN

<, >, <=, >=, =, <>, != (), NOT, AND, OR

RECURSIVE JOIN

LEFT OUTER JOIN

RIGHT OUTER JOIN

INSERT INTO (Attributes)
VALUES() / Select statement

UPDATE
SET
WHERE

DELETE FROM
WHERE

COMMIT

ROLLBACK

CREATE SCHEMA AUTHORIZATION,
DROP SCHEMA

CREATE TABLE (NOT NULL /
UNIQUE / DEFAULT / CHECK /
PRIMARY KEY / FOREIGN KEY),
ALTER TABLE (ADD / MODIFY),
DROP TABLE

CREATE TABLE AS

CREATE VIEW, DROP VIEW

CREATE INDEX, DROP INDEX

Numeric:
NUMBER(L,D)
INTEGER / INT
SMALLINT
DECIMAL(L,D)

Character:
CHAR(L)
VARCHAR(L) /
VARCHAR2(L)
Date: DATE