4

SQL: Data Definition

Prerequisites for This Section

- **Readings:**
  - **Required:** Connolly and Begg, sections 5.1 and 5.2
  - **Required:** Connolly and Begg, sections 6.1, 6.2 and 6.3

- **Assessments:**
  - Multiple-Choice Quiz 3

Section Objectives

In this section you will learn:

① Purpose and importance of SQL.

② Data types supported by SQL standard.

③ How to create and drop a database.

④ How to create, modify and drop a table.

⑤ How to create and drop an index.

DreamHome Rental Database

The relational schema for part of DreamHome case study is:

- **Branch** (branchNo, street, city, postcode)
- **Staff** (staffNo, fName, lName, position, sex, DOB, salary, branchNo)
- **PropertyForRent** (propertyNo, street, city, postcode, type, rooms, rent, ownerNo, staffNo, branchNo)
- **Client** (clientNo, fName, lName, telNo, prefType, maxRent)
- **PrivateOwner** (ownerNo, fName, lName, address, telNo)
- **Viewing** (clientNo, propertyNo, viewDate, comment)
- **Registration** (clientNo, branchNo, staffNo, dateJoined)

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

SQL is an Ideal Database Language

- An ISO standard now exists for SQL
- SQL is a language with four major components:
  - DDL (Data Define Language) for defining database structure.
  - DML (Data Manipulation Language) for populating the tables.
  - DQL (Data Query Language) for querying the tables
  - DCL (Data Control Language) for accessing control data.
- 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.

Characteristics of SQL

① Non-procedural

② Integrative with DDL, DML, DQL, DCL

③ Set-oriented operations

④ Be used in two ways (interactive and embedded), but same syntax

⑤ Simple and easy to learn

Functions of SQL

Functions	Statements
DDL	CREATE， DROP， ALTER
DML	INSERT， UPDATE， DELETE
DQL	SELECT
DCL	GRANT， REVOKE

History of SQL

- In 1974, D. Chamberlin (IBM San Jose Laboratory) defined language called ‘Structured English Query Language’ (SEQUEL).

- SQL is still pronounced ‘see-quel’, though official pronunciation is ‘S-Q-L’.

- IBM subsequently produced a prototype DBMS called *System R*, based on SEQUEL/2.

- Until SQL:1999, SQL did not contain flow of control commands.

History of SQL

- In late 70s, ORACLE appeared and was probably first commercial RDBMS based on SQL.
- In 1987, ANSI and ISO published an initial standard for SQL.
- In 1989, ISO published an addendum that defined an ‘Integrity Enhancement Feature’.
- In 1992, first major revision to ISO standard occurred, referred to as SQL2 or SQL/92.
- In 1999, SQL:1999 was released with support for object-oriented data management.
- In late 2003, SQL:2003 was released.

Importance of SQL

- SQL has become a Federal Information Processing Standard (FIPS), to which conformance is required for all sales of databases to American Government.

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

Reserved Words and User-defined Words

- 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.

Readability

- Most components of an SQL statement are *case insensitive*, except for literal character data.
- More readable with indentation and lineation:
  - 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, each should appear on a separate line and be indented under start of clause.

BNF Notation of SQL Syntax

- Use extended form of **BNF (Backus Naur Form) notation**:

  - **Upper-case** letters represent reserved words.

  - **Lower-case** letters represent user-defined words.

  - **Vertical Bar (|)** indicates a *choice* among alternatives.

  - **Curly brackets ({ })** indicate a *required element*.

  - **Square brackets ([ ])** indicate an *optional element*.

  - **Ellipsis (…)** indicates *optional repetition* (0 or more).

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).

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

Database Objects

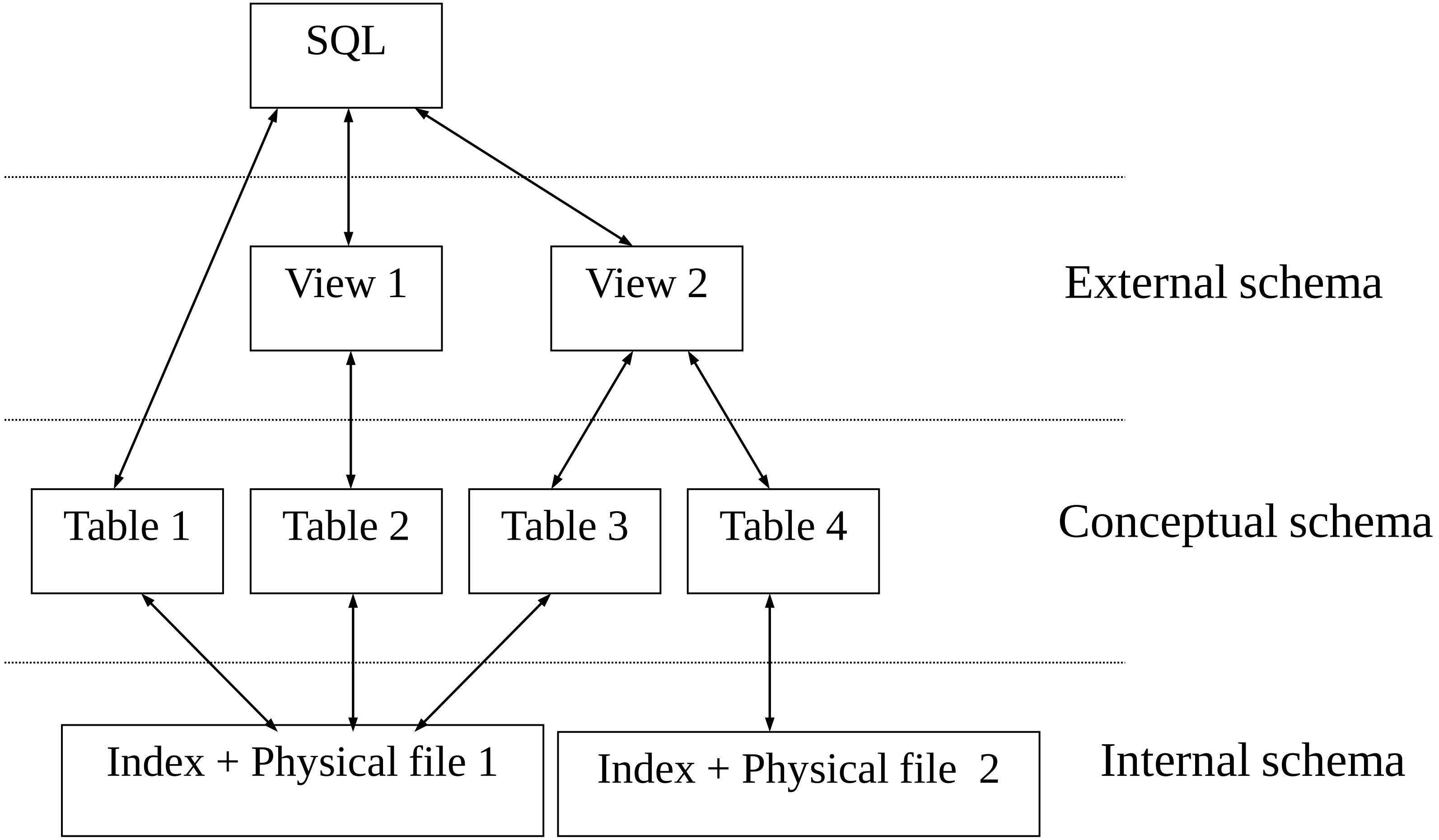
- Relations and other database objects exist in an ***environment***.
- Each environment contains one or more ***catalogs***, and each catalog consists of a set of ***schemas***.
- Schema
  - A schema is a named collection of database objects that are in some way related to one another
  - The objects in a schema can be tables, views, domains, and indexes.
  - All the objects in a schema have the same owner.

Main SQL DDL Statements

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

Functions Objects	Create	Modify	Drop
SCHEMA	CREATE SCHEMA		DROP SCHEMA
DOMAIN	CREATE DOMAIN	ALTER DOMAIN	DROP DOMAIN
TABLE	CREATE TABLE	ALTER TABLE	DROP TABLE
INDEX	CREATE INDEX		DROP INDEX
VIEW	CREATE VIEW		DROP VIEW

DDL Supports Three-level Schema



Creating a Database

- The process of creating a database differs significantly from product to product.
- In multi-user systems, the authority to create a database is usually reserved for the DBA. In a single-user system, a default database may be established when the system is installed.
- The schema definition statement syntax :

```
CREATE SCHEMA Name
    AUTHORIZATION CreatorIdentifier;
```

- Example:

```
CREATE SCHEMA DreamHome_db
    AUTHORIZATION Guoqing;
```

Dropping a Database

- The schema dropping statement syntax :

DROP SCHEMA database-name [**RESTRICT | CASCADE**];

  - **RESTRICT** is the **default**
  - 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:

DROP SCHEMA DreamHome_db **RESTRICT** ;

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

SQL Identifiers

- **SQL identifiers** are used to identify objects in the database, such as table names, view names, and columns.
- The **characters** of that can be used in a user-defined SQL identifier must appear in a character set.
- The **character set** consists of the upper-case letters A … Z, the lower-case letters a … z, the digits 0 … 9, and the underscore (_) character.
- The following restrictions are imposed on an identifier:

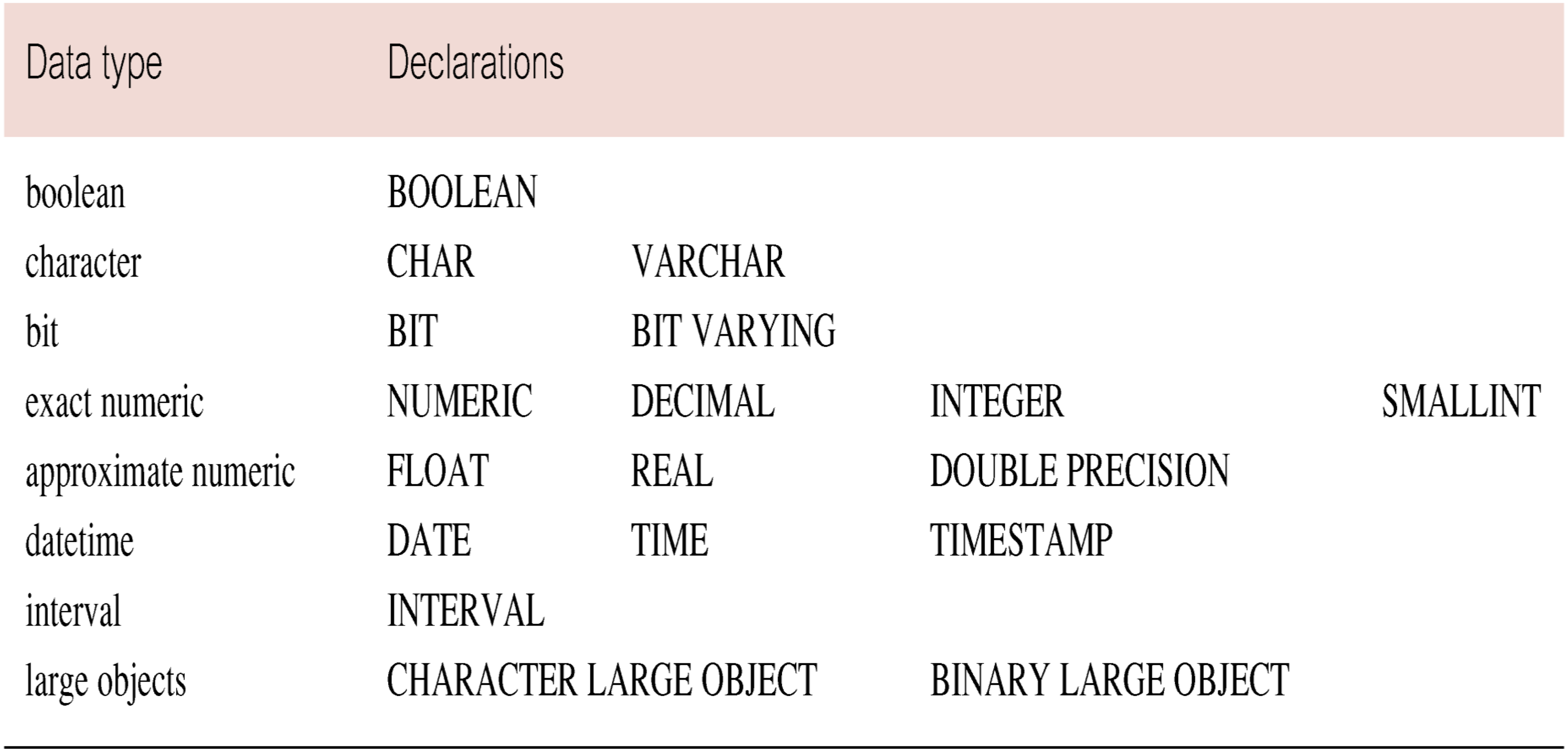
  ① an identifier can be no longer than 128 characters;

  ② an identifier must start with a letter;

  ③ an identifier cannot contain spaces.

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	

ISO SQL Data Types

① Boolean data

- consists of the distinct truth values TRUE, UNKNOWN, and FALSE.
- any comparison involving the NULL value or an UNKNOWN truth value returns an UNKNOWN result.

ISO SQL Data Types

② Character data

- The format for specifying a character data type is:

  CHAR [length]   or   **VARCHAR** [length]

- Example: branch **CHAR**(4),   address **VARCHAR** (30)

- A length can be specified to indicate the maximum number of characters that the column can hold (default length is 1).

- If the string is defined to be a **fixed length** and we enter a string with fewer characters than this length, the string is padded with blanks on the right to make up the required size.

- If the string is defined to be a **varying length** and we enter a string with fewer characters than this length, only those characters entered are stored, thereby using less space.

ISO SQL Data Types

③ Bit data

- The bit data is used to define bit strings, that is, a sequence of binary digits (bits), each having either the value 0 or 1.

- The format for specifying the bit data type is:

  BIT [VARYING] [length]

- Example:

  bitString **BIT** (4)

ISO SQL Data Types

④ Exact numeric data

- The exact numeric data type is used to define numbers with an exact representation.
- There are several ways of specifying an exact numeric data type:
  - ① **NUMERIC** [precision [, scale]]
  - ② **DECIMAL** [precision [, scale]]
  - ③ **INTEGER**
  - ④ **SMALLINT**
- **Examples:**
  - rooms **SMALLINT**
  - salary **DECIMAL**(7,2)

ISO SQL Data Types

⑤ Approximate numeric data

- The approximate numeric data type is used to define numbers that do not have an exact representation, such as real numbers.
- There are several ways of specifying an approximate numeric data type:

  ① **FLOAT** [precision]

  ② **REAL**

  ③ **DOUBLE PRECISION**

- Approximate numeric is similar to scientific notation in which a number is written as a mantissa times some power of ten (the exponent), for example, 10E3, +5.2E6, -0.2E-4.

ISO SQL Data Types

⑥ Datetime data

- The datetime data type is used to define points in time to a certain degree of accuracy.
- The ISO standard subdivides the datetime data type into YEAR, MONTH, DAY, HOUR, MINUTE, SECOND, TIMEZONE_HOUR, and TIMEZONE_MINUTE.
- Three types of datetime data type are supported:

① **DATE**

② **TIME** [timePrecision] **[WITH TIME ZONE]**

③ **TIMESTAMP** [timePrecision] **[WITH TIME ZONE]**

ISO SQL Data Types

⑦ Interval data

- The interval data type is used to represent periods of time.
- Every interval data type consists of a contiguous subset of the fields: **YEAR, MONTH, DAY, HOUR, MINUTE, SECOND.**
- Two classes of interval data type: **year-month** intervals and **daytime** intervals.
- The format for specifying the interval data type is:

INTERVAL {{startField TO endField} singleDatetimeField}

startField = **YEAR | MONTH | DAY | HOUR | MINUTE**
[(intervalLeadingFieldPrecision)]

endField = **YEAR | MONTH | DAY | HOUR | MINUTE | SECOND**
[(fractionalSecondsPrecision)]

singleDatetimeField = startField | **SECOND**
[(intervalLeadingFieldPrecision [, fractionalSecondsPrecision])]

ISO SQL Data Types

⑦ Interval data

- In all cases, startField has a leading field precision that defaults to 2.
- Example 1:
  - **INTERVAL YEAR(2) TO MONTH**
  - Represent an interval of time with a value between 0 years 0 months, and 99 years 11 months
- Example 2:
  - **INTERVAL HOUR TO SECOND(4)**
  - Represent an interval of time with a value between 0 hours 0 minutes 0 seconds, and 99 hours 59 minutes 59.0000seconds (the fractional precision of second is 4)

ISO SQL Data Types

⑧ Large objects

The SQL:2003 Standard defines both character large objects and binary large objects.

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

Creating a Domain

- The statement syntax :

CREATE DOMAIN DomainName **[AS]** dataType

[**DEFAULT** defaultOption]

[**CHECK** (searchCondition)]

- A domain is given a name, DomainName, a data type, an optional default value, and an optional CHECK constraint.
- Example:

```
CREATE DOMAIN SexType AS CHAR
        DEFAULT ‘M’
        CHECK (VALUE IN (‘M’, ‘F’));
        sex   SexType  NOT NULL；
```

Removing a Domain

- The statement syntax :

  DROP DOMAIN DomainName [**RESTRICT | CASCADE**];

- The drop behavior, **RESTRICT** or **CASCADE**, specifies the action to be taken if the domain is currently being used.
  - If **RESTRICT** is specified and the domain is used in an existing table, view, or assertion definition, the drop will fail.
  - In the case of **CASCADE:**

    ① any table column that is based on the domain is automatically changed to use the domain's underlying data type, and

    ② any constraint or default clause for the domain is replaced by a column constraint or column default clause, if appropriate.

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

Creating a Table

- The statement syntax:

```
CREATE TABLE TableName
    ({columnName 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)] [,…] });
```

Creating a Table

- The statement explanation:
  - 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.

Creating a Table

- Example 1

```
①  CREATE DOMAIN OwnerNumber AS VARCHAR(5)
        CHECK (VALUE IN (SELECT ownerNo FROM PrivateOwner));
②  CREATE DOMAIN StaffNumber AS VARCHAR(5)
        CHECK (VALUE IN (SELECT staffNo FROM Staff));
③  CREATE DOMAIN PNumber AS VARCHAR(5);
④  CREATE DOMAIN PRooms AS SMALLINT;
        CHECK(VALUE BETWEEN 1 AND 15);
⑤  CREATE DOMAIN PRent AS DECIMAL(6,2)
        CHECK(VALUE BETWEEN 0 AND 9999.99);.
```

Creating a Table

- Example 1

```
CREATE TABLE PropertyForRent (
    propertyNo   PNumber        NOT NULL, ….
    rooms        PRooms         NOT NULL         DEFAULT 4,
    rent         PRent          NOT NULL,        DEFAULT 600,
    ownerNo      OwnerNumber    NOT NULL,
    staffNo      StaffNumber    Constraint StaffNotHandlingTooMuch ….
    branchNo     BranchNumber   NOT NULL,
    PRIMARY KEY (propertyNo),
    FOREIGN KEY (staffNo) REFERENCES Staff
      ON DELETE SET NULL ON UPDATE CASCADE ….);
```

Altering a Table

- The statement syntax:

```
ALTER TABLE TableName

[ADD [COLUMN] columnName datatype [NOT NULL][UNIQUE]

[DEFAULT defaultOption][CHECK (SEARCHcONDITION)]]

[DROP [COLUMN] columnName [RESTRICT|CASCADE]]

[ADD[CONSTRAINT [constraintName]] tableConstraintDefinition]

[DROP CONSTRAINT constraintName [RESTRICT|CASCADE]]

[ALTER [COLUMN] SET DEFAULT defaultOption]

[ALTER [COLUMN] DROP DEFAULT];
```

Altering a Table

- The statement explanation:

① **Add** a new column to a table**.**

② **Drop** a column from a table.

③ **Add** a new table constraint.

④ **Drop** a table constraint.

⑤ **Set a default** for a column.

⑥ **Drop a default** for a column**.**

Altering a Table

- Example 1

Change Staff table by removing default of ‘Assistant’ for position column and setting default for sex column to female (‘F’).

```
ALTER TABLE Staff
        ALTER position DROP DEFAULT;
ALTER TABLE Staff
        ALTER sex SET DEFAULT ‘F’;
```

Altering a Table

- Example 2

Remove constraint from PropertyForRent that says staff are not allowed to handle more than 100 properties at a time. Add new column to Client table.

```
ALTER TABLE PropertyForRent
    DROP CONSTRAINT StaffNotHandlingTooMuch;
ALTER TABLE Client
    ADD prefNoRooms PRooms;
```

Dropping a Table

- The statement syntax :

  DROP TABLE TableName **[RESTRICT | CASCADE]**

- The statement explanation:
  - Removes named table and all rows within it.
  - 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).
- Example

  DROP TABLE PropertyForRent**;**

Agenda

1. **Introduction to SQL**
2. **Writing SQL Commands**
3. **Creating and Dropping a Database**
4. **SQL Data Types**
5. **Creating a Domain**
6. **Creating, Modifying, and Dropping a Table**
7. **Creating and Removing an Index**

Creating an Index

- The statement syntax:

  CREATE [UNIQUE] INDEX IndexName

  ON TableName **(**columnName **[ASC|DESC] [, …]);**

- The statement explanation:
  - **ASC |DESC** specify that the order is ascending or descending
  - **ASC** is the default setting
  - If the **UNIQUE** clause is used, uniqueness of the indexed column will be enforced by the DBMS.
- Example:

CREATE UNIQUE INDEX StaffNoInd **ON** Staff (staffNo);

Removing an Index

- The statement syntax :

  DROP INDEX IndexName;

- Example:

  DROP INDEX StaffNoInd;

Section Objectives

In this section you will learn:

① Purpose and importance of SQL.

② Data types supported by SQL standard.

③ How to create and drop a database.

④ How to create, modify and drop a table.

⑤ How to create and drop an index.

Questions?

Assignments

- **Exercise 3: part Ⅰ**

Prerequisites for Next Section

- **Readings:**
  - **Required:** Connolly and Begg, section 5.3
- **Assessments:**
  - Multiple-Choice Quiz 3