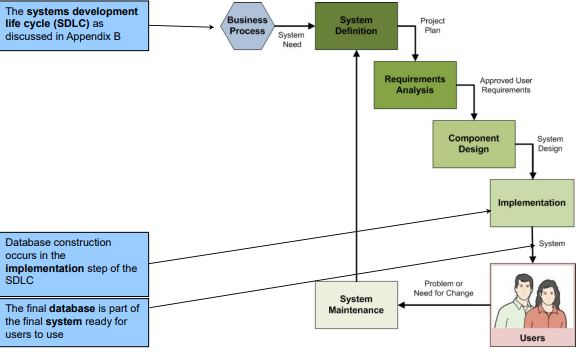
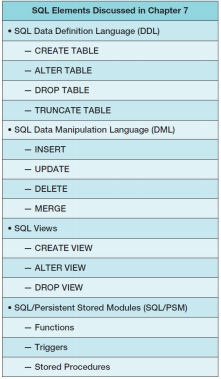
**Data Modeling in the SDLC**

**SQL Categories**

SQL statements can be divided into five categories:

* Data definition language (DDL)
* Data manipulation language (DML) statements
* SQL/Persistent Stored Modules (SQL/PSM) statements
* Transaction control language (TCL) statements
* Data control language (DCL) statements

Data definition language (DDL) statements

– Used for creating tables, relationships, and other structures

Data manipulation language (DML) statements

– Used for:

• Queries – SQL SELECT statement

• Inserting data – SQL INSERT statement

• Modifying data – SQL UPDATE statement

• Deleting data – SQL DELETE statement

SQL/Persistent Stored Modules (SQL/PSM) statements

– Add procedural programming capabilities

• Variables

• Control-of-flow statements – Covered in Chapters:

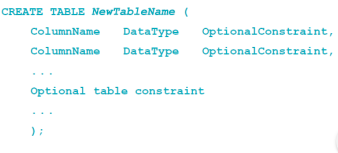
Transaction control language (TCL) statements

– Used to mark transaction boundaries and control transaction behavior

Data control language (DCL) statements

– Used to grant (or revoke) database permissions to (from) users and groups

**SQL CREATE TABLE Statement**

* CREATE TABLE statement is used for creating relations.
* Each column is described with three parts: column name, data type, and optional constraints.
* ****Format:

Constraints can be defined within the CREATE TABLE statement, or they can be added to the table after it is created using the ALTER table statement.

Column and table constraints include:

– PRIMARY KEY

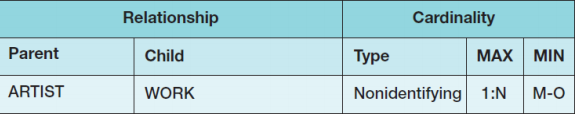
─ may not have NULL values

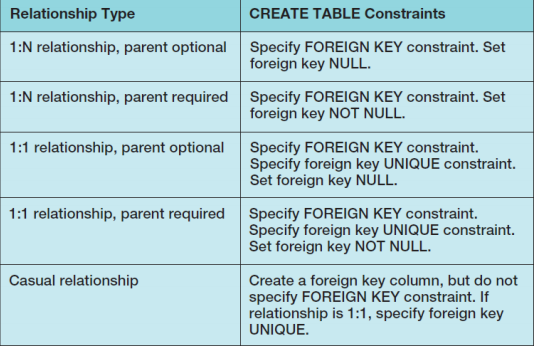
– FOREIGN KEY ─ may not have NULL values

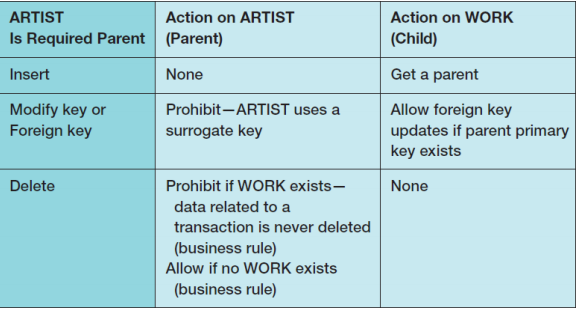
– NULL / NOT NULL – UNIQUE – CHECK

The DEFAULT keyword (not a constraint)

**Creating Relationships**

****

****

****

**Implementing Cardinalities**

|  |  |
| --- | --- |
| Relationship Type | CREATE TABLE Constraints |
| 1:N relationship, parent optional | Specify FOREIGN KEY constraint. Set foreign key NULL. |
| 1:N relationship, parent required | Specify FOREIGN KEY constraint. Set foreign key NOT NULL. |
| 1:1 relationship, parent optional | Specify FOREIGN KEY constraint.  SPECIFY foreign key UNIQUE constraint.  Set foreign key NULL. |
| 1:1 relationship, parent required | Specify FOREIGN KEY constraint.  Specify foreign key UNIQUE constraint.  Set foreign key NOT NULL |
| Casual relationship | Create a foreign key column, but do not specify FOREIGN KEY constraint. If relationship is 1:1, specify foreign key UNIQUE |

**SQL ALTER TABLE Statement**

• The **SQL ALTER TABLE statement** changes table structure, properties, or constraints after it has been created.

**Adding and Dropping Columns**

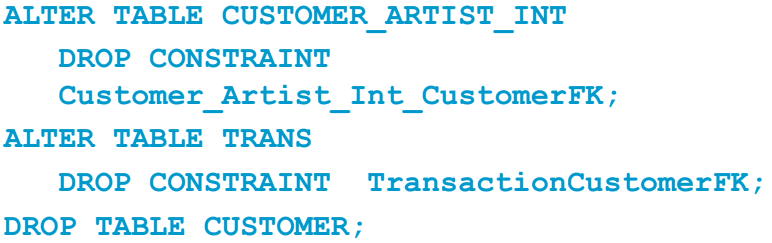
• The following statement will add a column named MyColumn to the CUSTOMER table:  
– Note that the **SQL COLUMN keyword** is ***not*** used!

• The **SQL ALTER TABLE statement** can be used to add a constraint

• The **SQL ALTER TABLE statement** can be used to drop a constraint:

**Removing Tables**

The **SQL DROP TABLE statement**: if there are constraints



**Removing Data Only**

• The **SQL TRUNCATE TABLE statement**:  
• Cannot be used with a table that is referenced by a foreign key constraint.  
• Resets surrogate key values to initial value.

**SQL DDL – CREATE INDEX**

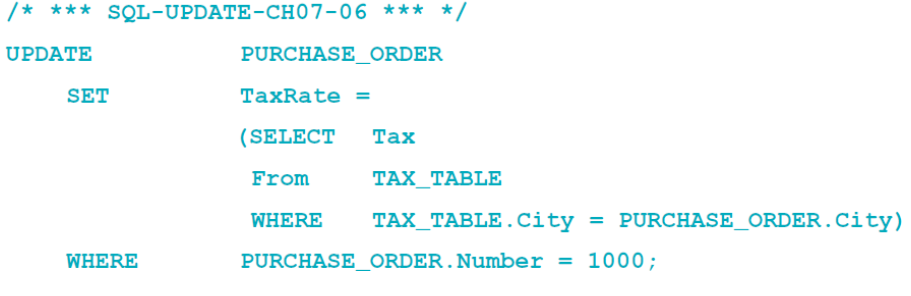
• An index is a data structure used to improve database performance.  
• The **SQL CREATE INDEX statement**• The **SQL ALTER INDEX statement**• The **SQL DROP INDEX statement**• See:

– Chapter 10A - Microsoft SQL Server 2014  
– Chapter 10B - Oracle Database  
– Chapter 10C - MySQL 5.6

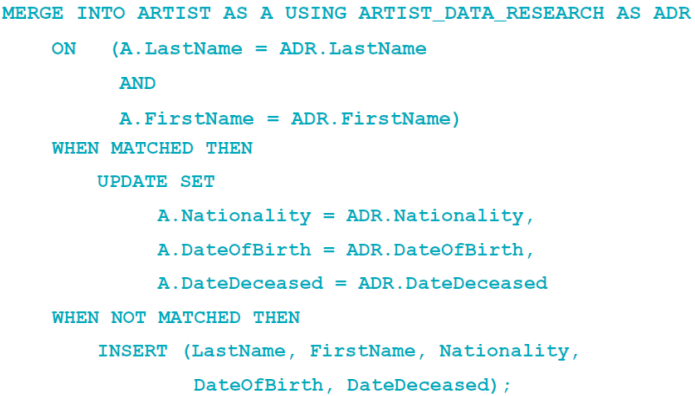
**Populating Tables**

• **Cannot** just use SQL INSERT statement by itself.  
• See discussions of how to handle this situation

• The **SQL UPDATE statement**:



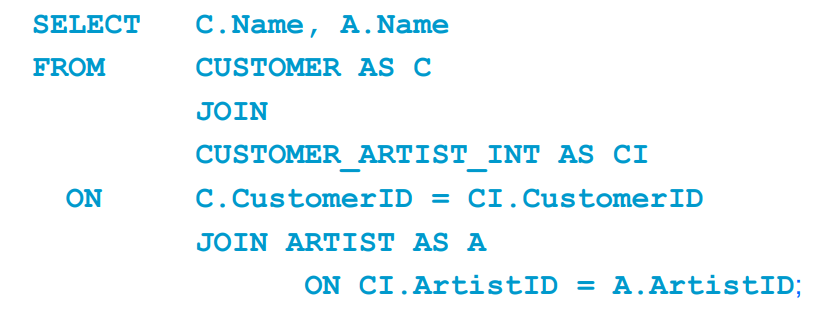
• The **SQL MERGE statement**:



• SQL DELETE statement:

• If you omit the **WHERE** clause, you will delete every row in the table.  
• Does ***not***reset surrogate key values.

• Use of aliases:



DBMS products differ



**SQL Views**

• An **SQL view** is a virtual table that is constructed from other tables or views.  
• It has no data of its own, but obtains data from tables or other views.  
• SELECT statements are used to define views:  
– A view definition may not include an ORDER BY clause.  
• SQL views are a subset of the external views:  
– They can be used only for external views that involve one multivalued path through the schema.

**SQL view uses**

|  |  |  |
| --- | --- | --- |
| **Hide columns or rows** | **Display results of computation** | **Hide complicated SQL syntax** |
| **Layer-built-in functions** | **Provide level of isolation between table data and users’ view of data** | **Assign different processing permissions to different views of same table** |
| **Assign different triggers to different views of the sam etable** | | |

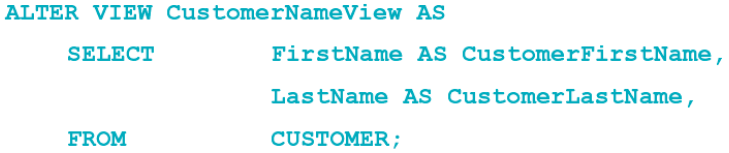
**SQL CREATE VIEW Statement**

• The **SQL CREATE VIEW statement**:  
• In the SQL standard, views do *not* support the **SQL ORDER BY clause**.  
– Individual DBMS products may support the SQL ORDER BY clause – see documentation.

• To see the results, use an **SQL SELECT statement** with the **view name** as the table name in the FROM clause:

**SQL ALTER VIEW Statement**

• The **SQL ALTER VIEW statement**:

  
• In the Oracle Database or MySQL 5.6, use the **SQL CREATE OR REPLACE VIEW statement**.  
– This allows creation and modification of SQL VIEW code.

|  |  |
| --- | --- |
| **Updatable Views** | **Possibly Updatable Views** |
| View based on a single table with no computed columns and all non-null oclumns present in the view | Based on a single table, primary key in view, some required columns missing from view, update and delete may be allowed. Insert is not allowed. |
| View based on any number of tables, with or without computed columns, and INSTEAD OF trigger defined for the view | Based on multiple tables, updates may be allowed on the most subordinate table in the view if rows of that table can be uniquely identified |

**Embedding SQL in Program Code**

• SQL cursors are used to select one row at a time from pseudo-files.  
• Problem: assigning SQL table columns with program variables

– Solution: object-oriented programming, PL/SQL  
• Problem: paradigm mismatch between SQL and application programming language:

– SQL statements return sets of rows; an application works on one row at a time  
– Solution: process the SQL results as pseudo-files

**SQL Cursors in Program Code**

• SQL can be embedded in triggers, stored procedures, and program code.

**SQL/Persistent Stored Modules (SSL/PSM)**

• **SQL/Persistent Stored Modules (SQL/PSM)** is an ANSI/ISO standard for embedding procedural programming functionality into SQL  
• Each DBMS product implements SQL/PSM in a different way, with some closer to the standard than others.  
– Microsoft SQL Server 2014 calls its version **Transact-SQL (T-SQL)**.  
– Oracle Database calls its variant **Procedural Language/SQL (PL/SQL)**.  
– MySQL 5.6 implements SQL/PSM, but has no special name for its variant of SQL.

**User-Defined Functions**

• A **user-defined function (stored function)** is a stored set of SQL statements that:  
– is *called by name* from another SQL statement  
– may have *input parameters* passed to it by the calling SQL statement, and  
– *returns an output value* to the SQL statement hat called the function

**Triggers**

A **trigger** is a stored program that is executed by the DBMS whenever a specified event occurs on a specified table or view.

Three trigger types:

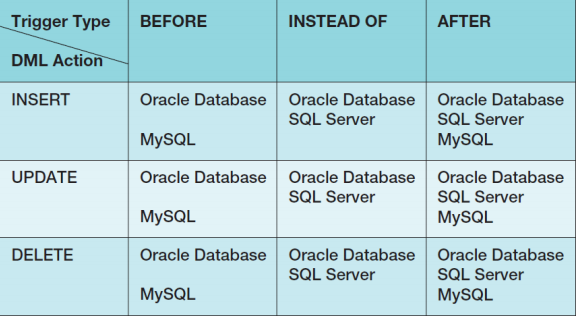
**BEFORE**, **INSTEAD** **OF**, and **AFTER**

– Each type can be declared for Insert, Update, and Delete.

– Resulting in a total of nine trigger types.

Oracle supports all nine trigger types.

SQL Server supports six trigger types (INSTEAD OF and AFTER).

****MySQL supports six trigger types (BEFORE and AFTER).

**Firing Triggers**

When a trigger is fired, the DBMS supplies:

– Old and new values for the update

– New values for inserts

– Old values for deletions

The way the values are supplied depends on the DBMS product.

Trigger applications include:

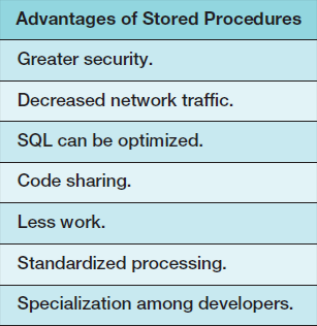
– Providing default values

– Enforcing data constraints

– Updating views

– Performing referential integrity actions

**Stored Procedures**

A **stored procedure** is a program that is stored within the database and is compiled when used.

– In Oracle, it can be written in PL/SQL or Java.

– In SQL Server, it can be written in TRANSACT-SQL.

Stored procedures can receive input parameters and they can return results.

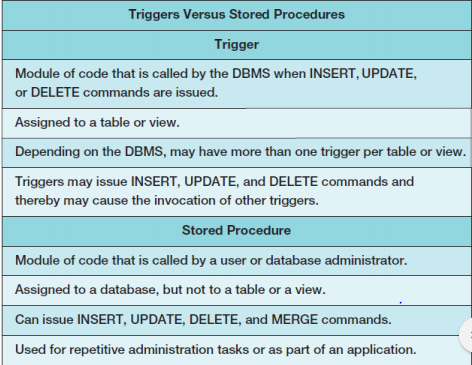
Stored procedures can be called from:

– Programs written in standard languages, e.g., Java, C#.

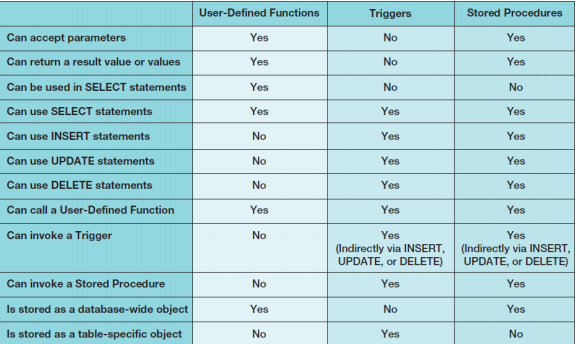
– Scripting languages, e.g., JavaScript, VBScript.

– SQL command prompt, e.g., SQL Plus, Query Analyzer.

**Triggers vs. Stored Procedures**

****

**Functions, Triggers, and Stored Procedures**

****