



Lesson 10: Basic SQL

SQL (Structure Query Language)

- Originally known as SEQUEL (Structured English Query Language) designed and implemented at IBM Research as the interface for an experimental relational database system called SYSTEMR. SQL is now the standard language for commercial relational DBMSs.
- SQL is a comprehensive database language
 - It has statements for data definitions, queries and updates.
- It is both DDL and a DML. It has facilities for defining views on the database for specifying security and authorization, for defining integrity constraints, and for specifying transaction controls. It also has rules for embedding SQL statements into a general-purpose programming language such as Java, COBOL, or C/C++.

SQL Data Definition and Data Types

- SQL uses the terms, table, row and column for the formal relational model terms *relation*, *tuple* and *attribute*.
- The main SQL command for data definition is the CREATE statement which can be used to create schemas, tables (relations), and domains.

Schema and Catalog Concepts in SQL

- Early versions of SQL did not include the concept of a relational database schema. The concept of an SQL schema was incorporated starting with SQL2 in order to group together tables and other constructs that belong to the same database application.
- An SQL schema is identified by a schema name, and includes an authorization identifier to indicate the user or account who owns the schema, as well as descriptors for each element in the schema.
- Schema elements include tables, constraints, views, domains and other constructs that describe the schema.
- A schema is created via CREATE SCHEMA statement, which can include all the schema elements and definitions.
- For example, the following statement create a schema called COMPANY, owned by the user with authorization identifier 'Jsmith'. Note each statement in SQL ends with a semicolon
CREATE SCHEMA COMPANY AUTHORIZATION 'Jsmith';

The CREATE TABLE in Command in SQL

- This command is used to specify a new relation by giving it a name and specifying its attributes and initial constraints. The attributes are specified first, and each attribute is given a name, a data type to specify its domain of values, and any attribute constraints, such as NOT NULL.
- The key, entity integrity and referential integrity constraints can be specified within the CREATE TABLE statement after the attributes are declared, or they can be added later using the ALTER TABLE command.

Attribute Data Types and Domains in SQL

The basic data types available for attributes include numeric, character string, bit string, Boolean, date, and time.

- Numeric data types include integer numbers of various sizes, and floating point numbers of various precision. Formatted numbers can be declared by using DECIMAL or NUMERIC.
- Character-string data type are either fixed length or CHARACTER. When specifying literal string value, it is placed between single quotation marks, and it is case sensitive.
- Bit-string data type are either of fixed length or varying length. Literal bit strings are placed between single quote but preceded by a B to distinguish them from character strings.
Example B'10101
- Boolean data type has the traditional values of TRUE or FALSE.
- Date data type has ten positions, and its components are YEAR, MONTH, and DAY in the form YYYY-MM-DD. The TIME data type has at least eight positions, with the components HOUR, MINUTE, and SECOND in the form HH:MM:SS.



LA IMMACULADA CONCEPCION SCHOOL
SENIOR HIGH SCHOOL
GRADE 12 – ICT: COMPUTER PROGRAMMING

Lesson 10: Basic SQL

Student Task:

Directions: answer the following questions. Write your answer in a sheet of yellow paper.

1. How do the relations (tables) in SQL differ from the relations defined formally? Discuss the other differences in terminology. Why does SQL allow duplicate tuples in a table or in a query result?
2. List the data types that are allowed for SQL attributes.
3. How does SQL allow implementation of the entity integrity and referential integrity constraints described in previous lesson? What about referential triggered actions?



Lesson 11: Basic SQL

Specifying Constraints in SQL

- This include key and referential integrity constraints, restrictions on attribute domains and NULLs and constraints on individual tuples within a relation.

Specifying Attribute constraints and attribute defaults

- Because SQL allows NULLs as attribute values, a constraint NOT NULL may be specified if NULL is not permitted for a particular attribute. This is always implicitly specified for the attributes that are part of the primary key of each relation, but it can be specified for any other attributes whose values are required not to be NULL, as shown in Figure 4.1. It is also possible to define a default value for an attribute by appending the clause DEFAULT <value> to an attribute definition. The default value is included in any new tuple if an explicit value is not provided for that attribute.
- The UNIQUE clause specifies alternate (secondary) keys, as illustrated in the DEPARTMENT and PROJECT table declarations. The UNIQUE clause can also be specified directly for a secondary key if the secondary key is a single attribute.

Giving Names to Constraints

- Following a keyword CONSTRAINT. The names of all constraints within a particular schema must be unique. A constraint name is used to identify a particular constraint in case the constraint must be dropped later and replaced with another constraint.

Specifying constraints on Tuples using CHECK

- In addition to key and referential integrity constraints, which are specified by special keywords, other table constraints can be specified through additional CHECK clauses at the end of a CREATE TABLE statement. These can be called tuple-based constraints because they apply to each tuple individually and are checked whenever a tuple is inserted or modified.
- For example, suppose that the DEPARTMENT table had an additional attribute Dept_create_date, which stores the date when the department was created. Then we could add the following CHECK clause at the end of the CREATE TABLE statement for the DEPARTMENT table to make sure that a manager's start date is later than the department creation date. CHECK (Dept_create_date <= Mgr_start_date); The CHECK clause can also be used to specify more general constraints using the CREATE ASSERTION statement of SQL.

Basic Retrieval Queries in SQL

- SQL has one basic statement for retrieving information from a database: the SELECT statement. The SELECT statement is not the same as the SELECT operation of relational algebra. There are many options and flavors to the SELECT statement in SQL, so we will introduce its features gradually.
- We present the features of SQL for simple retrieval queries. Features of SQL for specifying more complex retrieval queries are presented. Before proceeding, we must point out an important distinction between SQL and the formal relational model discussed in Previous lessons: SQL allows a table (relation) to have two or more tuples that are identical in all their attribute values. Hence, in general, an SQL table is not a set of tuples, because a set does not allow two identical members; rather it is a multiset (sometimes called a bag) of tuples. Some SQL relations are constrained to be sets because a key constraint has been declared or because the DISTINCT option has been used with the SELECT statement.

The SELECT-FROM-WHERE Structure of Basic SQL Queries

- Queries in SQL can be very complex. We will start with simple queries, and then progress to more complex ones in a step-by-step manner. The basic form of the SELECT statement, sometimes called a mapping or a select-from-where block.
-