M26CDE

Database Systems

Contents

- Introduction to
  - SQL
- SQL/Oracle: Oracle Data Types
- SQL Statements
  - Create tables
  - Delete tables
  - View tables created
  - Constraints
- Case Study: Airline Database

SQL

- SQL stands for Structured Query Language
- A standard for building and manipulating databases
- Can be used as
  - a stand alone query language
  - an embedded language: such as SQL statements embedded in, C++, Java, JSP (JavaServer Pages)
- Three main components:
  - DDL
  - DML
  - DCL

DDL and DML

- DDL (Data Definition Language)
  - For creating and manipulating the structure of the database
  - E.g., CREATE, DROP
- DML (Data Manipulation Language)
  - For populating and manipulating the data within the database
  - For essential database operations Create, Read, Update, and Delete (CRUD)
  - E.g., INSERT, SELECT
- Usually used by database administrators (DBA)

DCL

- DCL (Data Control Language)
  - GRANT - gives user's access privileges to database
  - REVOKE - withdraw access privileges given with the GRANT command

Oracle/SQL

- ORACLE Data Types : (more see Appendix 2)
  - **CHAR(n):** stores fixed length of character stings (n denotes number of characters specified), e.g. employees' ids.
  - **VARCHAR2(n):** stores variable length of character stings (n denotes number of characters specified), e.g. addresses.
  - **NUMBER(size, dec):** size is the maximum number of digits; dec is number of digits after the decimal point
  - **DATE:** stores date and time values

SQL – Create Tables (1)

- Create a table
  - **Syntax:**

    CREATE TABLE *table_name*
    (column_name1 data type constraint,
    column_name2 data type constraint,
    …)
  - **Example:** Student (s_id, s_name, date_of_birth)

    CREATE TABLE Student
    (s_id CHAR(6) PRIMARY KEY,
    s_name VARCHAR2(20) NOT NULL,
    date_of_birth DATE NOT NULL);

SQL – Create Tables (2)

- Create a table with a composite primary key
  - **Syntax:**

    CREATE TABLE *table_name*
    (column_name1 data type,
    column_name2 data type,
    …
    PRIMARY KEY *(column_name1, column_name2))*
  - **Example:** Person (<u>name</u>, <u>address</u>, date_of_birth)

    CREATE TABLE Person
    (name VARCHAR2(20),
    address VARCHAR2(50),
    date_of_birth DATE NOT NULL,
    PRIMARY KEY (name, address));

SQL – Create Tables (3)

- Create a table with Foreign Keys

  Syntax:

  CREATE TABLE *table_name*
  (column_1 data_type constraint,
  column_2 data_type REFERENCES *parent_table,*
  …)

  where column_2 is the foreign key.

SQL – Create Tables (4)

- Example: Student (s_id, s_name, *c_id)
  Course (course_id, title)

```
CREATE TABLE Course
(course_id     CHAR(5)       PRIMARY KEY,
 title         VARCHAR2(30)  NOT NULL);

CREATE TABLE Student
(s_id    CHAR(6)       PRIMARY KEY,
s_name VARCHAR2(30) NOT NULL,
c_id     CHAR(5)       REFERENCES  Course);
```

SQL – Create Tables (5)

The tables can also be created with the following statements:

```
CREATE TABLE Course
(course_id        CHAR(5) ,
 title            VARCHAR2(30)        NOT NULL
 CONSTRAINT pk_course PRIMARY KEY(course_id)
 );

CREATE TABLE Student
(s_id      CHAR(6)          PRIMARY KEY,
s_name VARCHAR2(30) NOT NULL,
c_id       CHAR(5),
CONSTRAINT fk_cid  FOREIGN KEY(c_id)
                   REFERENCES Course(course_id)
);
```

SQL – Removing Tables

- They are two statements for removing a table from the database:

  DROP TABLE *table_name;*

  Example:

  DROP TABLE Student;

  If a table is dropped, all the relationships with other tables will no longer be valid, the integrity constraints will be dropped, grant or access privileges on the table will also be dropped.

ALTER TABLE statement

- To add a column in a table, use the following syntax:
  - ALTER TABLE table_name
    ADD column_name datatype
    - Add a new column
  - E.g ALTER TABLE Course ADD department VARCHAR2(15)
- To delete a column in a table, use the following syntax:
  - ALTER TABLE table_name
    DROP COLUMN column_name
    - Remove a column
  - E.g. ALTER TABLE Course DROP COLUMN department;
- To change the data type of a column in a table:
  - ALTER TABLE table_name
    MODIFY column_name datatype
    - Modify column

  - ALTER TABLE Course MODIFY department VARCHAR2(20);

SQL – View Tables

- View a list of tables created
  - **SQL Statement:**

    SELECT TABLE_NAME FROM USER_TABLES;

    (Note: You need to type in the above statement word by word.)
- View the details of a table created
  - **Syntax:**

    DESC *table_name*
  - **Example**

    DESC Student

Constraints

Common types of constraint include the following:

- **NOT NULL** Constraint: Ensures that a column cannot have NULL value.
- **DEFAULT** Constraint: Provides a default value for a column when none is specified.
- **UNIQUE** Constraint: Ensures that all values in a column are different.
- **CHECK** Constraint: Makes sure that all values in a column satisfy certain criteria.
- **Primary Key** Constraint: Used to uniquely identify a row in the table.
- **Foreign Key** Constraint: Used to ensure referential integrity of the data.

Constraints

```
CREATE TABLE Student
    (s_id  INTEGER CHECK (s_id>0),
     s_name    VARCHAR2(20) NOT NULL,
     date_of_birth        DATE    NOT NULL,
     PRIMARY KEY s_id);

CREATE TABLE Student
    (s_id  INTEGER PRIMARY KEY ,
     s_name    VARCHAR2(20) NOT NULL,
     date_of_birth        DATE    NOT NULL,
     credits DEFAULT 0);
```

- INSERT INTO Student (s_id, s_name, date_of_birth) values (‘8’,‘Kim’, ’12/10/1980’);
- Credits not specified.

Adding a Constraint

Consider the following tables:

```
CREATE TABLE Course
(course_id          CHAR(5) ,
 title              VARCHAR2(30)   NOT NULL
 CONSTRAINT pk_course PRIMARY KEY(course_id)
);

CREATE TABLE Student
(s_id       CHAR(6)          PRIMARY KEY,
s_name      VARCHAR2(30) NOT NULL,
c_id        CHAR(5),
);
```

To add a foreign key constraint so that c_id refers to course_id :

```
ALTER TABLE ADD CONSTRAINT fk_cid  FOREIGN KEY(c_id)
                                   REFERENCES Course(course_id);
```

Case Study: Airline Database

- Basis of all SQL examples and exercises for this module
  - Design and table definitions of the database are given in Appendix 1.
- Practical work involved:
  - Create the database using Oracle
  - Populate the database
  - Manipulate the data