Database Systems

SQL: Data Definition language (DDL)

Topics

- Essential syntax to use DDL – data definition language
- Data types supported by SQL standard
- **How to define tables and modify its structure**
- **How to define integrity constraints in SQL**

Data Definition Language

- SQL DDL allows database objects such as schemas, domains, tables, views, and indexes to be created and deleted
- Main SQL DDL statements are:
  - CREATE SCHEMA — DROP SCHEMA
  - CREATE/ALTER DOMAIN — DROP DOMAIN
  - **CREATE/ALTER TABLE** — **DROP TABLE**
  - **CREATE VIEW** — **DROP VIEW**
- Many DBMSs also provide:
  - **CREATE INDEX** — **DROP INDEX**
  - CREATE DATABASE — DROP DATABASE

Data Definition Language (DDL)

- Tables and other database objects exist in an environment
- Each environment contains >=1 catalogs, and each catalog consists of set of schemas
- **Schema is named collection of related database objects with the same owner**
- **Objects in a schema can be tables, views, security, domains, assertions, collations, translations, and character sets**

DROP TABLE

- `DROP TABLE TableName`
  `[ RESTRICT | CASCADE [CONSTRAINTS] ]`
- `e.g. DROP TABLE PropertyForRent;`
- 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)
- Note Oracle's diff. : CASCADE CONSTRAINTS

CREATE TABLE

```
CREATE TABLE PropertyForRent
   (propNo   VARCHAR2(5) NOT NULL    UNIQUE
   , ...
   ,rooms    NUMBER(2,0) DEFAULT 4   NOT NULL
   ,rent     NUMBER(6,2) DEFAULT 600 NOT NULL
             CHECK(rent BETWEEN 0 AND 9999.99)
   ,ownerNo  VARCHAR2(5) NOT NULL
   ,staffNo  VARCHAR2(5) NOT NULL
   ,branchNo CHAR(4)     NOT NULL
   ,PRIMARY KEY (propNo)
   ,FOREIGN KEY (staffNo) REFERENCES Staff (staffNo)
             ON DELETE SET NULL ON UPDATE CASCADE
   , ...
   ,CONSTRAINT chkrms CHECK(rooms BETWEEN 1 AND 15)
    ...);
```

SQL Data Types

CATEGORY	STORES THESE DATA
Character string	Strings of characters
Binary large object	Binary data
Exact numeric	Integers and decimal numbers
Approximate numeric	Floating-point numbers
Boolean	Truth values: true, false, or unknown
Datetime	Date and time values
Interval	Date and time intervals

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	

- `CHAR(9)`
- `VARCHAR(30)` `VARCHAR2(30)`
- `DECIMAL(10,2)` `NUMBER(10,2)`
- `DATE` `(Oracle variant)`

SQL Data Types

Table 3.3. Examples of Literals

LITERAL	EXAMPLES
Character string	`'42', 'ennui', 'don''t', N'Jack'`
Numeric	`42, 12.34, 2., .001, -123, +6.33333, 2.5E2, 5E-3`
Boolean	`trUE, FALSE, UNKNOWN`
Datetime	`DATE '2005-06-22', TIME '09:45:00', TIMESTAMP '2006-10-19 10:23:54'`
Interval	`INTERVAL '15-3' YEAR TO MONTH, INTERVAL '22:06:5.5' HOUR`

CREATE TABLE

- Creates a table with one or more columns of the specified data type
- 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 actions

CREATE TABLE – partial syntax

```
CREATE TABLE TableName (
  {colName dataType [NOT NULL] [UNIQUE]
          [DEFAULT defaultOption]
          [CHECK (searchCondition)]
   [,...]}
  [,PRIMARY KEY (listOfColumns)]
 {[,UNIQUE (listOfColumns)] [,...]}
 {[,FOREIGN KEY (listOfFKColumns)
    REFERENCES ParentTable [(listOfCKColumns)]
          [ON UPDATE referentialAction]
          [ON DELETE referentialAction]]
   [,...]}
  {[,CHECK (searchCondition)] [,...] })
```


CREATE TABLE

```
CREATE TABLE PropertyForRent
   (propNo   VARCHAR2(5) NOT NULL    UNIQUE
   , ...
   ,rooms    NUMBER(4,0) DEFAULT 4   NOT NULL
   ,rent     NUMBER(6,2) DEFAULT 600 NOT NULL
             CHECK(rent BETWEEN 0 AND 9999.99)
   ,ownerNo  VARCHAR2(5) NOT NULL
   ,staffNo  VARCHAR2(5) NOT NULL
   ,branchNo CHAR(4)     NOT NULL
   ,PRIMARY KEY (propNo)
   ,FOREIGN KEY (staffNo) REFERENCES Staff (staffNo)
             ON DELETE SET NULL ON UPDATE CASCADE
   , ...
   ,CONSTRAINT chkrms CHECK(rooms BETWEEN 1 AND 15)
    ...);
```

Integrity Constraints

- Consider 5 types of integrity constraints:
  - Required data: **NOT NULL**

    position VARCHAR2(10) NOT NULL
  - Domain constraints
  - **Entity integrity – Primary Key**
  - **Referential integrity – Foreign Key**
  - Enterprise constraints
    - **CHECK constraints**
    - Triggers
- **CONSTRAINT** keyword – only if naming the constraint

Entity Integrity

- Primary key of a table must contain a unique, non-null value for each row.
- ISO standard supports PRIMARY KEY clause in CREATE and ALTER TABLE statements:
  - PRIMARY KEY(staffNo)
  - PRIMARY KEY(clientNo, propertyNo)
- Can only have one PRIMARY KEY clause per table; can still ensure uniqueness for alternate keys using UNIQUE:
  - UNIQUE(telNo)

Referential Integrity

- FK is column or set of columns that relates each row in child table containing FK to row of parent table with matching PK or unique key
- Referential integrity means that, if FK contains a value, that value must reference an existing row in parent table
- Definition of FKs with FOREIGN KEY clause in CREATE and ALTER TABLE:
  - CONSTRAINT fk_branchNo FOREIGN KEY(branchNo) REFERENCES Branch
  - FOREIGN KEY(orderNo, prodNo) REFERENCES OrderDetail (orderNo, productNo)

Referential Integrity

- Any INSERT/UPDATE that attempts to create FK value in child table without matching PK or unique key value in parent will be rejected
- Action taken that attempts to update/delete a PK or unique key value in parent table with matching rows in child is dependent on referential action specified using ON UPDATE and ON DELETE subclauses:
  - CASCADE
  - NO ACTION
  - SET DEFAULT
  - SET NULL
- Note: Oracle's limited support in this

Referential Integrity

- NO ACTION: Reject delete from parent - Default
- CASCADE: Delete row from parent and delete matching rows in child, and so on in cascading manner
- SET NULL: Delete row from parent and set FK column(s) in child to NULL; only valid if FK columns are defined as NULL allowed
- SET DEFAULT: Delete row from parent and set each component of FK in child to specified default; only valid if DEFAULT value is specified for each FK column involved

CHECK Constraints

- **[CONSTRAINT constraint_name] CHECK (condition);**
- Minimum or maximum values

```
CONSTRAINT chk_rent
CHECK(rent <= 9999.99)
```

- Specific list of values

```
CONSTRAINT chk_branchNo
CHECK(branchNo IN ('BR09','BR22','BR17'))
```

- Range of values

```
CONSTRAINT chk_rent
CHECK(rent BETWEEN 0 AND 9999.99)
```

ALTER TABLE

- Add a new column to a table
- Drop a column from a table
- Add a new table constraint
- Drop a table constraint
- Set a default for a column
- Drop a default for a column

ALTER TABLE – Partial Syntax

```
ALTER TABLE table
{ [ALTER COLUMN column_name
    { new_data_type [ ( precision [ , scale ] ) ]
        [ NULL | NOT NULL ]
        }
  ]
  | ADD
      { [ < column_definition > ]
      |  column_name AS computed_column_expression
      } [ ,...n ]
  | [ WITH CHECK | WITH NOCHECK ] ADD
      { < table_constraint > } [ ,...n ]
  | DROP
      { [ CONSTRAINT ] constraint_name
          | COLUMN column } [ ,...n ]
  | { CHECK | NOCHECK } CONSTRAINT
      { ALL | constraint_name [ ,...n ] }
  | { ENABLE | DISABLE } TRIGGER
      { ALL | trigger_name [ ,...n ] }
}
```


ALTER TABLE

```
ALTER TABLE MyTable
    ADD column_b VARCHAR(20) NULL;
```

```
ALTER TABLE MyTable
    DROP COLUMN column_b;
```

```
ALTER TABLE MyTable
    ADD
      column_b VARCHAR(20) NULL
      CONSTRAINT exb_unique UNIQUE;
```

```
ALTER TABLE MyTable WITH NOCHECK
    ADD
    CONSTRAINT exa_check
    CHECK (column_a > 1);
```

ALTER TABLE

```
ALTER TABLE MyTable
  /* Add a PRIMARY KEY identify column */
     ADD column_b NUMBER(5) NOT NULL
     CONSTRAINT column_b_pk PRIMARY KEY,
  -- Add a column referencing another
  --    column in the same table
          column_c NUMBER(5)     NULL
           CONSTRAINT column_c_fk
           REFERENCES MyTable(column_b);
```

- Use multiple ALTER TABLE statements for readability

DROP TABLE

```
DROP TABLE TableName
[  RESTRICT | CASCADE [CONSTRAINTS] ]
```

- e.g. `DROP TABLE PropertyForRent;`
- 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)
- Note Oracle's diff. : CASCADE CONSTRAINTS