Introduction to Operating Systems and SQL for Data Science

Practice 7 - Relational Model

Previously on the lecture:

- Database background
- Relational Model (tables)
- Relations (tables)
- Attributes (Fields, Columns)
- Records (Tuples)
- Keys (Primary key, Foreign key)
- Constraints (Key, Entity Integrity, Referential Integrity)

Scheme definition

- Each table has a name, for example: Students
- The table has Attributes.
- The name of the table and its attributes together define the schema of the relation or table (Relation Schema), and the marking of this is:
  - R (a1,…., An)

Example in this case:

Students (FirstName, LastName, **<u>ID</u>**, YearOfBirth, City)

The key is marked by an underline.

Reichman University

Attribute value type

- We will define each attribute in the table its type and domain of values (Domain).

- Common types:
  - Integer
  - Decimal
  - Character
  - Boolean
  - VARCHAR (String)

Reichman University

Example for defining attributes

Age = {Integer; Age>0 and Age<120}

First_Name = {VARCHAR (20)}

Keys

Key - one (or more) fields which allows the distinction between records and the unambiguous identification of each record in relation.

We distinguish between the different keys:

Super Key - A subset of fields (non-minimal) that identify a record.

Candidate Key - A subset of fields (minimum) that identify a record

Primary Key - One of the candidate keys.

Foreign Key - A field (fields) that refers to the key (Primary Key) of another relation (base relation).

That is: the field value exists in the base relational key. (foreign key name does not have to be the same as the corresponding primary key name but must be from the same domain.)

Reichman University

Integrity constraints

They were designed to ensure data integrity within and between relationships.

- **Key constraint** – a relation has a key.
- **Entity Integrity constraint** - The Primary Key must contain a value (not null).
- **Referential Integrity constraint** - Each value of a "foreign" key in a table must have the same value in its base table key or Null.

Foreign Key and Primary Key are defined in the same domain. Foreign Key values are a subset of the Primary Key values.

Reichman University

Check constraint

- Check is used to verify the data.
- for example:

  Grade smallint check (value between 0 and 100)

Reichman University

Keys

- Key can be a single attribute (for example, plantID in plant table) or several attributes.

- We can define primary and foreign keys (containing two or more fields)
  CONSTRAINT fk_pk FOREIGN KEY(plantID, fieldID) references Other_Table(plantID, fieldID) ON UPDATE CASCADE ON DELETE no action;

- when defining a foreign key we can specify what happens with the entries that are related to the foreign key:
  - NO ACTION - can't delete or update the entry with the key on the parent_table if there are entries that depends on this key on child tables.

Reichman University

Attributes

- An attribute can hold an empty value, unless **explictly** specified.

  to explicit specify that an attribute can’t be null, we need to use the not null constraint.

Altering tables

- We can change existing tables using alter statements.
- for example, adding a constraint:

```
ALTER TABLE Persons
ADD CONSTRAINT PK_Person PRIMARY KEY (ID,LastName);
```

- adding a column:

```
ALTER TABLE table_name
ADD column_name datatype;
```

Reichman University

Datatypes

Existing datatypes (depends on the SQL implementation, there are several)

- datetime
- text
- float
- int,
- smallint
- decimal
- numeric
- BOOLEAN

Exercise (1)

The scheme below depicts a plant nursery. There are several fields that provide the nursery with plants.

Plants

[plantID, name, main_supplier, sub_supplier, country, family, bloom_season, cost, price]

Customers

[customerID, name, address, discount, credit_days]

Sales

[deal_no, plantID, customerID, amount, deal_date, pay_date]

Fields

[fieldID, region, main_grower, phoneNo, area]

FieldPlants

[plantID, fieldID, seed_date, bloom_date, pick_date]

Exercise (1)

Write **Data Definition statements** that creates the database **scheme**

- for every attribute (column) in a relation, determine:
    - value domain (value range)
    - default value (if needed)
    - constraints (referential integrity constraints, column contraints, etc..)

- Relations between entities.

Reichman University

Exercise (1)

Plants

[plantID**,** name, main_supplier, sub_supplier, country, family, bloom_season, cost, price**]**

```
CREATE TABLE Plants
(

) ;
```

Exercise (1)

Plants

[plantID**,** name, main_supplier, sub_supplier, country, family, bloom_season, cost, price**]**

```
CREATE TABLE Plants
(
        plantID integer primary key,
        Name varchar(50) not null unique,
        Main_supplier varchar (50),
        sub_supplier varchar (50) default 'none',
        Country varchar (25) default 'Israel',
        family varchar (25) not null,
        bloom_season varchar(6),
        cost integer check(cost>=0),
        price integer check(price>0)
) ;
```

Exercise (1)

Customers

[customerID**,** name, address, discount, credit_days**]**

```
CREATE TABLE  Customers
(



);
```

Reichman University

Exercise (1)

Customers

[customerID**,** name, address, discount, credit_days**]**

```
CREATE TABLE  Customers
(
        customerID integer primary key,
        Name varchar(50) not null,
        address varchar (50),
        discount integer default 0 check (discount>=0),
        credit_days numeric default 0 check (credit_days >=0)
);
```

Reichman University

Exercise (1)

Sales

[deal_no, plantID, customerID, amount, deal_date, pay_date]

```
CREATE TABLE  Sales
(

);
```

Exercise (1)

Sales

[deal_no, plantID, customerID, amount, deal_date, pay_date]

```
CREATE TABLE  Sales
(
        deal_no integer primary key,
        plantID integer foreign key references Plants(plantID) ON
        UPDATE CASCADE ON DELETE no action,
        customerID integer foreign key references Customers
        (customerID) ON UPDATE CASCADE ON DELETE no action,
        amount  integer  not null default 1 check(amount>=0),
        deal_date datetime default getdate(),
        pay_date datetime default getdate() ,
        CONSTRAINT ch_pd check (pay_date >= deal_date )
);
```

Exercise (1)

Fields

[fieldID, region, main_grower, phoneNo, area]

```
CREATE TABLE  Fields
(




);
```

Reichman University

Exercise (1)

Fields

[fieldID, region, main_grower, phoneNo, area]

```
CREATE TABLE  Fields
(
        fieldID integer primary key,
        region varchar(50) not null,
        main_grower varchar(50),
        phoneNo varchar(11),
        area integer check (area >0)
);
```

Reichman University

Exercise (1)

FieldPlants

[plantID, fieldID, seed_date, bloom_date, pick_date]

```
CREATE TABLE  FieldPlants
(

);
```

Reichman University

Exercise (1)

FieldPlants

[plantID, fieldID, seed_date, bloom_date, pick_date]

```
CREATE TABLE  FieldPlants
(
        plantID integer foreign key references Plants(plantID) ON
        UPDATE CASCADE ON DELETE no action,
        fieldID integer foreign key references Fields(fieldD) ON
        UPDATE CASCADE ON DELETE no action,
        seed_date  datetime ,
        bloom_date datetime,
        pick_date datetime,
        CONSTRAINT tab_pk PRIMARY KEY(plantID , fieldD),
        CONSTRAINT ch_sd check(bloom_date >= seed_date),
        CONSTRAINT ch_bd check(pick_date >= bloom_date)
);
```

Reichman University