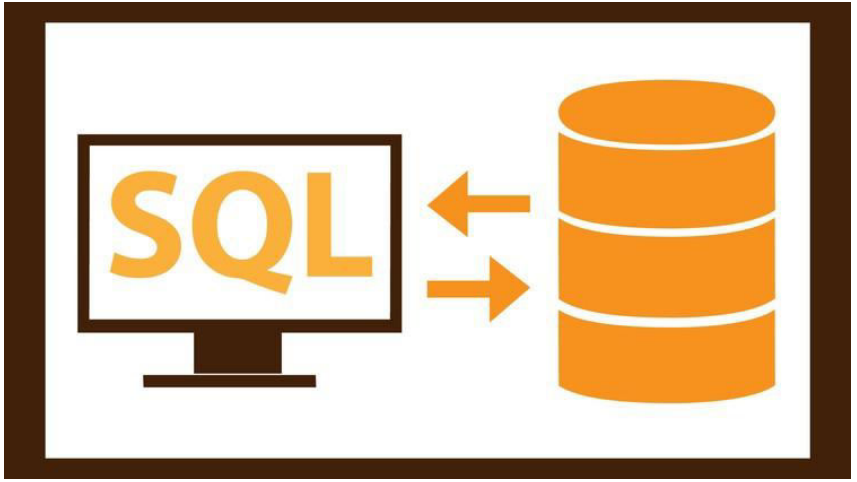


SQL-Intro

Index

- Databases as tables
- Create/delete tables
- Insert/delete data (in tables)
- Query Database

Databases as tables

How the programer sees (relational) databases

Students:

SSN	Name	Category
123-45-6789	Charles	undergrad
234-56-7890	Dan	grad
	…	…

Takes:

SSN	CID
123-45-6789	CSE444
	…

Courses:

CID	Name	Quarter
CSE444	Databases	fall
CSE541	Operating systems	winter

Informal Relational DataBase Design

- Describe the “world” as a set of “things”
- Each thing will be a table
- Thing are described by the table column
- Thing are related
- Some Relations are described by tables
- Others by an extra column

Example-1: Enroll in a activity

Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

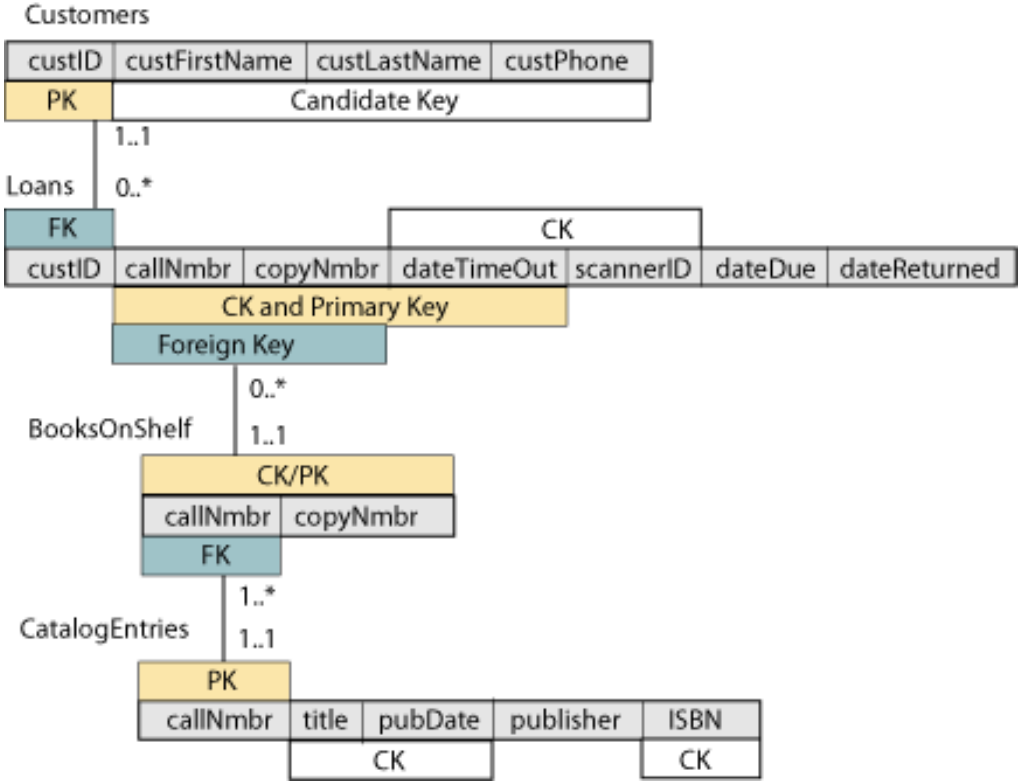
Participants Table

ID*	Activity*
084	Tennis
084	Swimming
100	Squash
100	Swimming
182	Tennis
219	Golf
219	Swimming
219	Squash

Activities Table

Activity*	Cost
Golf	$47
Sailing	$50
Squash	$40
Swimming	$15
Tennis	$36

Example-2: Library



Design Database Assignments

A friend is interested in keeping track of information about his album collection. He is not concerned about whether or not the albums are CDs, tapes, LPs, etc. Also, assume that he does not have any compilation albums—that is, each album has songs from a single band. For each album, he wants to store which band recorded the album, the title, the year, and the chronology (e.g. this is the 4th album for that band). He also wants to store the songs, including title, length, track number, and writer(s). Of course, if two bands record the same song, they might have different track numbers and lengths.

For each band (group or individual), he also wants to store the names of all of the band members. For each band member, he needs their first and last names, and country of origin. Consider both band members and songwriters as musicians.

chess tournament

wikiloc

SQL

DDL/DML/DCL

- **DDL stands from Data Definition Language:**
  - CREATE – to create objects in the database
  - ALTER – alters the structure of the database
  - DROP – delete objects from the database
  - GRANT – gives user's access privileges to database
  - REVOKE – withdraw access privileges given with the GRANT command
- **DML stands from Data Manipulation Language statements. Some examples:**
  - SELECT – retrieve data from the a database
  - INSERT – insert data into a table
  - UPDATE – updates existing data within a table
  - DELETE – deletes all records from a table, the space for the records remain
  - EXPLAIN PLAN – explain access path to data
  - LOCK TABLE – control concurrency
- **DCL stands from Data Control Language statements. Some examples:**
  - COMMIT – save work done
  - SAVEPOINT – identify a point in a transaction to which you can later roll back
  - ROLLBACK – restore database to original since the last COMMIT
  - SET TRANSACTION – Change transaction options like what rollback segment to use

SQL Types

Table 6.1 ISO 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	

Bool states, char(3) vs varchar, exact vs approximate, data vs interval

Create Table Construct

- An SQL relation is defined using the **create table** command:

  create table r (A_1 D_1, A_2 D_2, ..., A_n D_n,
  (integrity-constraint$_1$),
  ...,
  (integrity-constraint$_k$))

  - r is the name of the relation
  - each A_i is an attribute name in the schema of relation r
  - D_i is the data type of values in the domain of attribute A_i

Create Table Construct

- Example:

create table *branch*
(*branch-name* char(15) **not null,**
branch-city char(30),
assets integer)

Integrity Constraints in Create Table

- **not null**
- **primary key** (A_1, ..., A_n)
- **check** *(P),* where *P* is a predicate

Example: Declare *branch-name* as the primary key for *branch* and ensure that the values of *assets* are non-negative.

```
create table branch
    (branch-name char(15),
    branch-city     char(30)
    assets          integer,
    primary key (branch-name),
    check (assets >= 0))
```

Integrity Constraints in Create Table-II

- **unique**
- **foreign key** (A_1)
- **default** *value*
- **index**

Example:

```
CREATE TABLE Orders (
 OrderID int PRIMARY KEY,
 OrderNumber int NOT NULL UNIQUE,
 PersonID int REFERENCES Persons(PersonID) );
```

Drop and Alter Table Constructs

- The **drop table** command deletes all information about the dropped relation from the database.
- The **alter table** command is used to add attributes to an existing relation.

  alter table *r* **add** *A D*

  where *A* is the name of the attribute to be added to relation *r* and *D* is the domain of *A.*
  - All tuples in the relation are assigned *null* as the value for the new attribute.
- The **alter table** command can also be used to drop attributes of a relation

  alter table *r* **drop** *A*

  where *A* is the name of an attribute of relation *r*
  - Dropping of attributes not supported by many databases

Let us Go live

DDL examples previous chapter

The END