Databases and Information Systems CS303

Introduction to SQL
07-09-2023

Recap

- Relational Databases
  - Relations (Tables), Attributes (Columns)
- Keys
  - Primary Keys
  - Foreign Keys
- Schema representation

Relational Query Languages

- Procedural Languages: Instruct the system to perform a sequence of operations on the database to compute the desired result.

- Nonprocedural language: User describes the desired information without giving a specific procedure for obtaining that information.

- SQL: Mix of both.
  Most commonly used relational query language

SQL

- Developed by IBM in 1970s
- Originally called Sequel
- SQL : Structured Query Language
- Part of ISO standardization
- Many commercially available softwares support SQL
  PostGreSQL, MySQL, Oracle ...

What can SQL be used for?

- Data Definition Language: SQL provides commands for defining relation schemas, deleting relations, and modifying relation schemas.

- Data-manipulation language: SQL provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.

- Integrity: SQL includes commands for specifying integrity constraints that the data stored in the database must satisfy. Updates that violate integrity constraints are disallowed.

What can SQL be used for?

- View definition: SQL includes commands for defining views.
- Transaction control: SQL includes commands for specifying the beginning and ending of transactions.
- Embedded SQL and dynamic SQL: SQL statements can be embedded within general-purpose programming languages, such as Python, C, C++, and Java.
- Authorization: SQL commands for specifying access rights to relations and views.

Creating Tables

- Specify data types of the attributes

- Basic data types:
  - char(n): A fixed-length character string with user-specified length n.
  - varchar(n): A variable-length character string with user-specified maximum length n.

  - int: An integer (a finite subset of the integers that is machine dependent).
  - smallint: A small integer (a machine-dependent subset of the integer type).

  - numeric(p, d): A fixed-point number with user-specified precision. The number consists of p digits (plus a sign), and d of the p digits are to the right of the decimal point.
    Example: numeric(3,1) allows 44.5 to be stored exactly, but not 444.5 or 0.32

  - real, double precision: Floating-point and double-precision floating-point numbers with machine-dependent precision.
  - float(n): A floating-point number, with precision of at least n digits.

Data types related to Dates

- date: A calendar date containing a (four-digit) year, month, and day of the month.
- time: The time of day, in hours, minutes, and seconds.
- timestamp: A combination of date and time.

Creating of Databases

- Same tool can handle multiple databases:
  - University database
  - Company database
  - .....

- CREATEDB <db_name>;

- DROPDB <db_name>;
  [Deletes the database along with all relations of the database]

Creating of Relations

- CREATE TABLE department
  (dept_name varchar (20),
  building varchar (15),
  budget numeric (12,2),
  primary key (dept name));

- CREATE TABLE r
  (A1 D1 ,
  A2 D2 ,
  ...,
  An Dn ,
  integrity-constraint1 ,
  ...,
  integrity-constraintk);

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

department

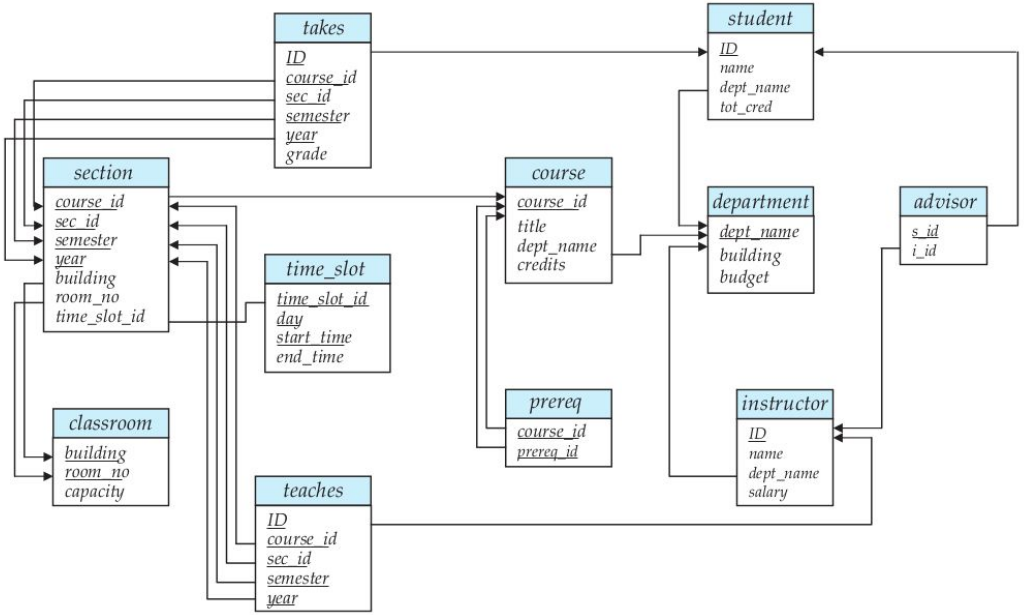
Various Integrity Constraints

- primary key (Aj1 , Aj2 , . . . , Ajm)
  - The attributes Aj1 , Aj2 , . . . , Ajm form the primary key for the relation.
  - The primary key attributes are required to be nonnull and unique
  - It is generally a good idea to specify a primary key for each relation.

- foreign key (Ak1 , Ak2 , . . . , Akn) references s
  - Specifies that the values of attributes (Ak1 , Ak2 , . . . , Akn) for any tuple in the relation must correspond to values of the primary key attributes of some tuple in relation s.

- not null
  - Specifies that the null value is not allowed for that attribute

Creating relations of University database

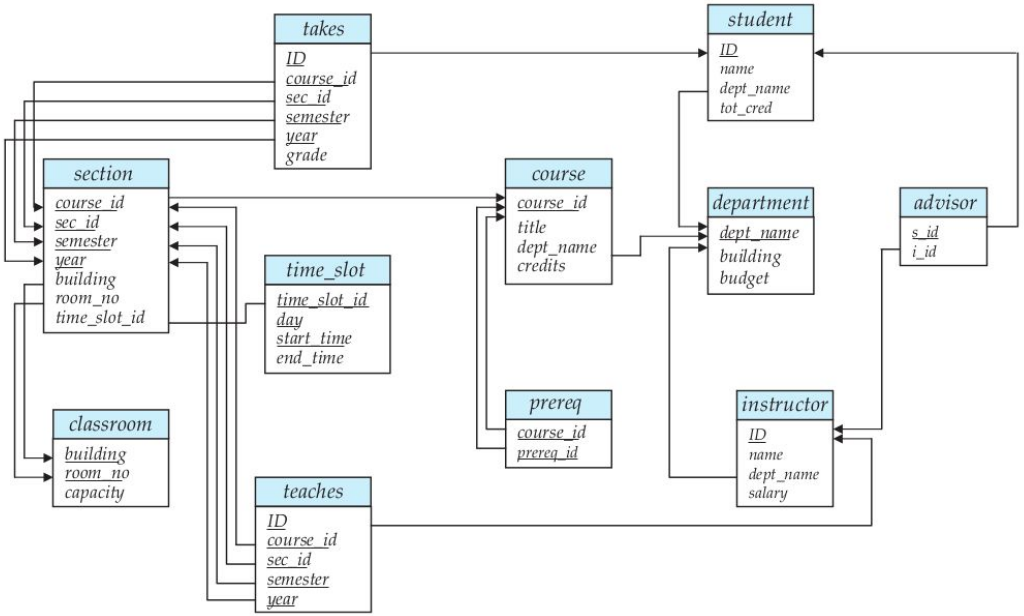



```
create table department
    (dept_name      varchar (20),
     building       varchar (15),
     budget         numeric (12,2),
     primary key (dept_name));

create table course
    (course_id      varchar (7),
     title          varchar (50),
     dept_name      varchar (20),
     credits        numeric (2,0),
     primary key (course_id),
     foreign key (dept_name) references department);

create table instructor
    (ID             varchar (5),
     name           varchar (20) not null,
     dept_name      varchar (20),
     salary         numeric (8,2),
     primary key (ID),
     foreign key (dept_name) references department);
```

Creating relations of University database



create table *section*
(*course_id* **varchar** (8),
sec_id **varchar** (8),
semester **varchar** (6),
year **numeric** (4,0),
building **varchar** (15),
room_number **varchar** (7),
time_slot_id **varchar** (4),
primary key (*course_id*, *sec_id*, *semester*, *year*),
foreign key (*course_id*) **references** *course*);

create table *teaches*
(*ID* **varchar** (5),
course_id **varchar** (8),
sec_id **varchar** (8),
semester **varchar** (6),
year **numeric** (4,0),
primary key (*ID*, *course_id*, *sec_id*, *semester*, *year*),
foreign key (*course_id*, *sec_id*, *semester*, *year*) **references** *section*,
foreign key (*ID*) **references** *instructor*);

Deleting a Table

- DROP <table_name>;

  Deletes the table and its content (Table disappears from the database)

- DELETE FROM <table_name>;

  Deletes the contents of the table but keeps empty table in the database (Empty table is still present in the database)

Altering a Table

- ALTER TABLE <table_name> ADD <new_column> <data_type>;
- ALTER TABLE <table_name> DROP <column_name>;

dept_name	*building*	*budget*
Biology	Watson	90000
Comp. Sci.	Taylor	100000
Elec. Eng.	Taylor	85000
Finance	Painter	120000
History	Painter	50000
Music	Packard	80000
Physics	Watson	70000

department

Inserting rows to a table

- INSERT INTO course
  VALUES
  (‘CS-303, ‘Database Systems’, ‘Comp. Sci.’, 6);

- INSERT INTO course (course_id, title, dept name, credits)
  VALUES (‘CS-437’, ’Database Systems’, ’Comp. Sci.’, 4);

- INSERT INTO course (title, course_id, credits, dept name) VALUES (’Database Systems’, ’CS-437’, 4, ‘Comp. Sci.’);

- INSERT INTO course (title, course_id, credits)
  VALUES (’Database Systems’, ’CS-437’, 4);

course_id	*title*	*dept_name*	*credits*
BIO-101	Intro. to Biology	Biology	4
BIO-301	Genetics	Biology	4
BIO-399	Computational Biology	Biology	3
CS-101	Intro. to Computer Science	Comp. Sci.	4
CS-190	Game Design	Comp. Sci.	4
CS-315	Robotics	Comp. Sci.	3
CS-319	Image Processing	Comp. Sci.	3
CS-347	Database System Concepts	Comp. Sci.	3
EE-181	Intro. to Digital Systems	Elec. Eng.	3
FIN-201	Investment Banking	Finance	3
HIS-351	World History	History	3
MU-199	Music Video Production	Music	3
PHY-101	Physical Principles	Physics	4

department

Inserting rows to a table

- INSERT INTO student
  values (‘3003’, ‘Green’, ’Finance’, null);

- INSERT INTO student (id, name, dept name)
  VALUES (‘3003’, ‘Green’, ‘Finance’);

ID	*name*	*dept_name*	*tot_cred*
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

student

SELECT queries

Basic Structure

- SELECT <list_of_attributes>
  FROM <tables(s)>
  WHERE <conditions> ;

- SELECT name
  FROM instructor;

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

instructor

name
Srinivasan
Wu
Mozart
Einstein
El Said
Gold
Katz
Califieri
Singh
Crick
Brandt
Kim

Avoid Duplicates

- SELECT dept_name
  FROM department ;

dept_name
Comp. Sci.
Finance
Music
Physics
History
Physics
Comp. Sci.
History
Finance
Biology
Comp. Sci.
Elec. Eng.

- SELECT DISTINCT dept_name
  FROM department ;

- Explicitly say we want all values (with duplicates)
  SELECT ALL dept_name
  FROM department;

ID	name	dept_name	salary
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

department

Mathematical Operations

- SELECT id, name, dept_name, salary*1.1
  FROM instructor;

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

instructor

WHERE clause

- Retrieve the names of all instructors of Computer Science department who have salary > 7000
- SELECT name
  FROM instructor
  WHERE dept_name = ‘Comp. Sci.’ AND salary > 70000

ID	*name*	*dept_name*	*salary*
10101	Srinivasan	Comp. Sci.	65000
12121	Wu	Finance	90000
15151	Mozart	Music	40000
22222	Einstein	Physics	95000
32343	El Said	History	60000
33456	Gold	Physics	87000
45565	Katz	Comp. Sci.	75000
58583	Califieri	History	62000
76543	Singh	Finance	80000
76766	Crick	Biology	72000
83821	Brandt	Comp. Sci.	92000
98345	Kim	Elec. Eng.	80000

instructor

name
Katz
Brandt