COSC210
Database Mangement Sytems

Lecture 3 - The Relational Data Model and Relational Database Constraints

Dr. Mitchell Welch / Dr. Edmund Sadgrove

Reading

- Chapter 5 from ***Fundamentals of Database Systems*** by Elmazri and Navathe

---

Summary

- Relational Data Model Concepts
- Relational Model Constraints
- DB Operations and Integrity Contraints (SQL)

Relational Data Model Concepts

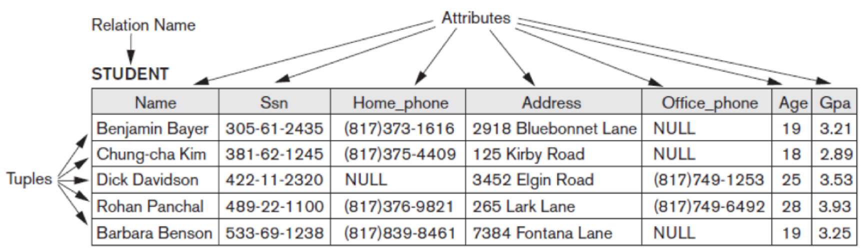
- The ***Relational Data Model*** represents a series of **Relations**.
- A **Relation** is a table:
    - **Row**: record or instance of the real-world.
    - **column**: attribute or cell of data.
    - Depicts relationships.

```
id,    name,    team,
1,     Amy,     Blues,
2,     Bob,     Reds,
3,     Chuck,   Blues,
4,     Richard, Blues,
5,     Ethel,   Reds,
6,     Fred,    Blues,
7,     Gilly,   Blues,
8,     Hank,    Reds,
9,     Hank,    Blues
```

* Differs from a flat-file (delimitted).

Relational Data Model Concepts

- The Relational Data-model uses its own **terminology**:
    - Table = *Relation*
    - Row = *Tuple*
    - Column = *Attribute*



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	(817)373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	(817)375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	(817)749-1253	25	3.53
Rohan Panchal	489-22-1100	(817)376-9821	265 Lark Lane	(817)749-6492	28	3.93
Barbara Benson	533-69-1238	(817)839-8461	7384 Fontana Lane	NULL	19	3.25

Figure 3.1
The attributes and tuples of a relation STUDENT.

Relational Data Model Concepts

- **Attributes** have a ***Domain***.
    - A *Domain* (*D*) is a set of atomic values (indivisible).
    - Specified by ***datatype***, ***atomicity*** and ***context*** (name).
- Examples:

- Phone numbers (local vs national).
- Academic (schools vs departments)

Relational Data Model Concepts

- The Relational Schema is the DB structure.
- A **Relation Schema** specifies:
  - The Relation **name**
  - A list of **attributes** of degree / arity (n)
- Formally: $R(A_1, A_2, \ldots A_n)$
  - Where R is the relation name, with each attribute A_i
- Further our domain: $dom(A_i)$
  - Where A_i is the i^{th} attribute in R.

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Relational Data Model Concepts

- A **Relation State** specifies:
  - A list of tuples.
  - Recall database state.
- Formally: $r(R)$
  - Where $r = \{ t_1, t_2, \ldots t_m \}$ and r is a set of m tuples.
- Each tuple (t) consists of values: ($t = < v_1, v_2, \ldots v_n >$)
  - Where each value is an element of **dom(A_i)**

* A formal defintion is useful in database design.

Relational Data Model Concepts

- A set of **Tuples** in $r(R)$ have **no order**.
- Ordering is **not part** of the formal defintion of $r(R)$.
- However, most implementations will have a **physical ordering**.
- Individual tuples are an ordered sequence of values:
  - In practice only need to match the order of attributes.
  - Tuples are therefore a mapping of
    - (<attribute>, <value>) pairs.

Student Table A

Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21
Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25

Student Table B

Name	SSN	HomePhone	Address	OfficePhone	Age	GPA
Dick Davidson	422-11-2320	null	3452 Elgin Road	749-1253	25	3.53
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	null	19	3.25
Charles Cooper	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Katherine Ashly	381-62-1245	375-4409	125 Kirby Road	null	18	2.89
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	null	19	3.21

Relational Data Model Concepts

- **PostgreSQL** will typically return things in the **physical order** in which they are stored.
  - Reorganizing the data structure will change this.
    - E.g. level of Indexing.
  - We should **specify order** (**if** ordering is **important**).
  - This will also make SQL *more* **portable**.

- Efficient (common queries vs storage order).

Relational Data Model Concepts

- Examples:

```
-- Attribute order not specified in the insert
INSERT INTO my_table
    VALUES (val_1,val_2 ... val_n);

-- Attribute order not specified in the insert
INSERT INTO my_table(attribute_1, attribute_2 ... attribute_n)
    VALUES (val_1,val_2 ... val_n);

-- Attribute order not specified in the insert
INSERT INTO employee(Fname,Lname,Dno,Ssn)
    VALUES ('Richard', 'Marini', 4, '5635633867');

-- Adjust the ordering of a B-tree index
CREATE INDEX test_desc_index ON test_table (id DESC NULLS LAST);
```

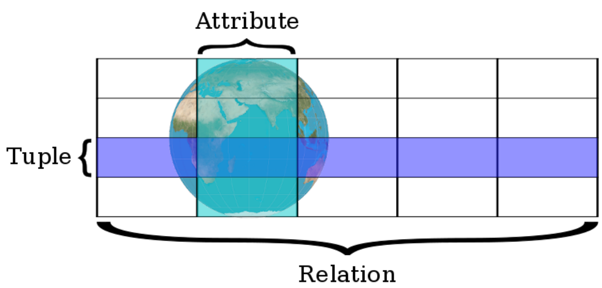
Relational Data Model Concepts

- Explicit ordering of tuples returned using the `ORDERBY` clause

```
SELECT d.dname, e.lname, e.fname, p.pname
FROM department d, employee e, works_on w, project p
WHERE d.dnumber=e.dno AND e.ssn = w.essn AND
      w.pno = p.pnumber
ORDER BY d.dname, e.lname, e.fname;
```

Relational Data Model Concepts

- The **Relation Schema** can be thought of as a **declaration** (or assertion).
  - Each tuple is a ***Fact***.
- The Relation Schema is a ***predicate***.
  - A logical statement (truth valued).
  - Values must therefore satisfy the predicate
- This is important when considering constraints on our DB.



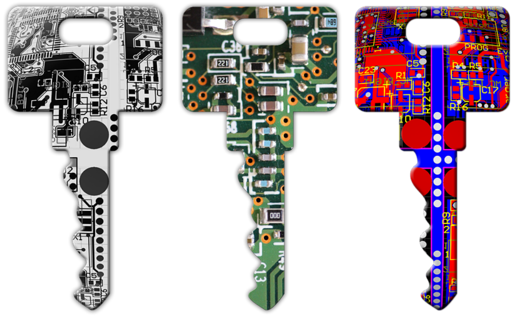
* The integrity of our database.

Relational Model Constraints

- In a **relational database** there will typically be many relations.
- Data integrity is enforced by contraints:
  - Rules that govern the relationships between tuples and relations.
- These fall into three high level categories:
  - **Model-based** Constraints (e.g. duplicates)
  - **Schema-based** Constraints (e.g. DDL)
  - **Application-based** constraints (e.g. business rules)

Relational Model Constraints

- A **Relation** is composed of a ***set* of tuples**
- The uniqueness property requires a **superkey**:
  - A subset of distinct values in a tuple.
  - Relations will have one default superkey.
  - Can have redundant values.



Relational Model Constraints

- The concept of a **key** is more useful.
- A *key* must be:
  - **Unique** for all tuples in each attribute of the key.
  - It must be a **minimal superkey**:
    - should not be able to remove any attributes and still satisfy the uniqueness property
- From this definition it is evident that a *key* is a *superkey* but not *vise versa*
  - A superkey does not need to be minimal.

Relational Model Constraints

- A **key**:
  - **Uniquely identifies** a tuple within the relation.
  - ***Time-invariant*** : new values inserted must be unique.
- **Candidate key**:
  - A relation may have multiple unique attributes.
  - These will be candidate keys.
- **Primary key**:
  - Chosen subset or single attribute value.

* E.g. serial number.

Relational Model Constraints

- **Entity Integrity Constraint**:
  - States **no primary key** value can be **NULL**.
    - Null values cannot be determined.
    - Therefore not distiquishable (not unique).
- **Referential Integrity Constraint**:
  - **Maintains consistency** between tuples in related tables (relations).
  - **Foreign key**:
    - Refers to a primary key in another relation(s).
    - The primary key must exist.
    - E.g. student table and grades table.

Relational Model Constraints

- A foreign key (FK) of R_1 references relation R_2 as long as:
  - **Same domain**:
    - Attributes in FK of R_1 match the domain of the primary key in R_2.
  - **Same value**:
    - Tuple t_1 of state $r_1(R_1)$ occurs in PK for some tuple t_2 in $r_2(R_2)$.
    - Specifically t_1[FK] = t_2[PK] or t_1[FK] is NULL.
  - **References**:
    - t_1 in $r_1(R_1)$ **references** t_2 in $r_2(R_2)$

Relational Model Constraints

- Referential integrity constraints arise from the relationships among entities.
- **Primary keys** cannot be NULL.
- **Foreign keys** can be NULL.
  - E.g. EMPLOYEEs without a department.
- A foreign key may reference its own relation.

Figure 3.6
One possible database state for the COMPANY relational database schema.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Relational Model Constraints

- Primary keys can be foreign keys as composite keys.

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

Relational Model Constraints

- Referential integrity constraints can be represented diagrammatically:

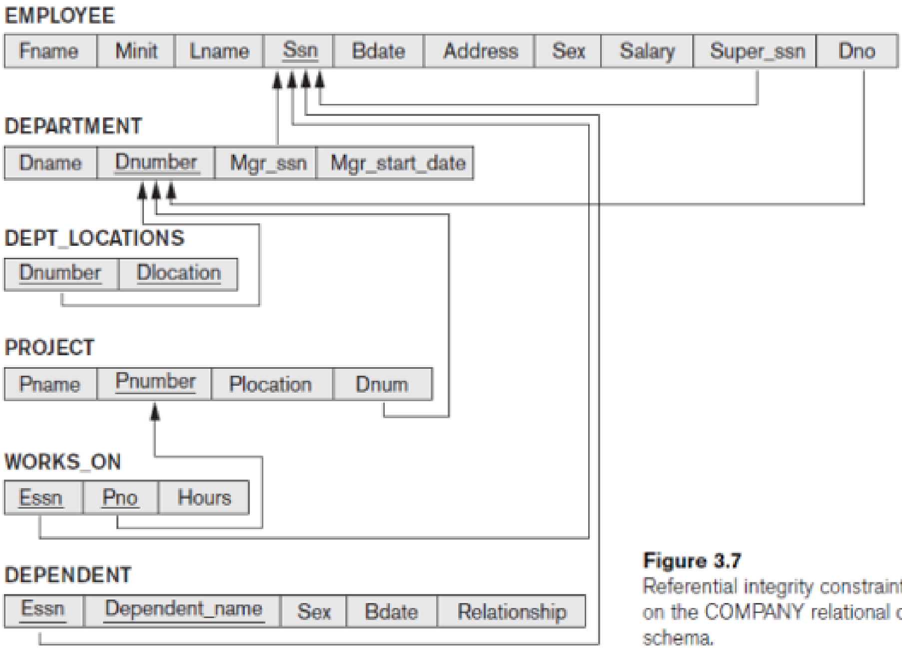
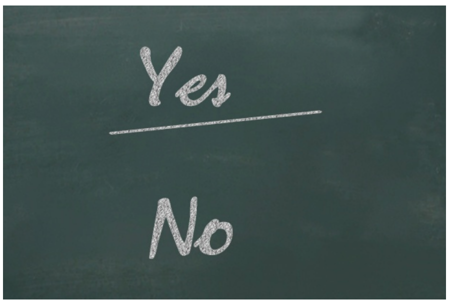


Figure 3.7
Referential integrity constraints displayed on the COMPANY relational database schema.

DB Operations and Referential Integrity

- **Operations** of the relational model take the form of either ***retrievals*** **and** ***updates***
- In this section we will **focus on updates**:
  - Insertion of new data (**INSERT**)
  - Removal of data (**DELETE**)
  - Changes to existing values (**UPDATE**)
- These operations should not violate **integrity constraints**.

DB Operations and Referential Integrity



- The **INSERT** operation:
  - A list of attribute values for a new tuple *t* in *r(R)*.
  - **Domain constraints**:
    - If attribute values do not lie within the attribute.
  - **Key constraints**:
    - Violated if a key value in the new tuple already exists.
  - **Entity integrity**:
    - Violated if a NULL value is provided for a primary key.
  - **Referential integrity**:
    - Violated if a foreign key does not exist within the referenced relation.
- If constraints are violated, the default operation is to **reject** the Insert.

DB Operations and Referential Integrity

- The **DELETE** operation:
  - Removes a specified tuple from *t* in *r(R)*.
  - **Referential integrity**:
    - Violated if feriegn key references deleted primary key.

- Adjustment operations needed:
  - ***Restrict***:
    - Rejects the deletion.
  - ***Cascade***:
    - Removes tuples from referencing relations.
  - ***Set Null***:
    - Sets referencing tuple foreign keys to NULL.

DB Operations and Referential Integrity

- The **UPDATE** operation:
  - Modifies values within tuple *t* in *r(R)*.
  - **Referential** and **Entity Integrity Constraints**:
    - Violated if update effects reference, not unique or NULL.
  - Adjustment operations needed:
    - ***Restrict***:
      - Rejects the update.
    - ***Cascade***:
      - Updates tuples in referencing relations.
    - ***Set Null***:
      - Sets referencing tuple foreign keys to NULL.

DB Operations and Referential Integrity

- **Actions** include:
  - Retrieval.
  - Update.
  - Delete.
- These can be **grouped** into one **transaction**.
  - An atomic unit of work.
- These transactions must leave the database in a **valid state**.
  - No constraint violations.

Referential Integrity Constraints in PostgreSQL

- No Referential action specified.
- The default action is to restrict any such update, insertion or deletion.

```
CREATE TABLE department (
  dname        varchar(25) not null,
  dnumber      integer primary key,
  mgrssn       char(9) not null,
  mgrstartdate date
);


CREATE TABLE project (
  pname      varchar(25) unique not null,
  pnumber    integer primary key,
  plocation  varchar(15),
  dnum       integer not null,
  foreign key (dnum) references department(dnumber)
```

```
);
```

Referential Integrity Constraints in PostgreSQL

- Referential action specified, with a self-referencing table.

```
CREATE TABLE employee (
  fname    varchar(15) NOT NULL,
  minit    varchar(1),
  lname    varchar(15) NOT NULL,
  ssn      char(9) PRIMARY KEY,
  bdate    date,
  address  varchar(50),
  sex      char,
  salary   decimal(10,2),
  superssn char(9),
  dno      integer,
  foreign key (superssn) references employee(ssn)
        ON DELETE SET NULL ON UPDATE CASCADE,
  foreign key (dno) references department(dnumber)
        ON DELETE SET NULL ON UPDATE CASCADE
);
```

Relational Model Constraints

- The **constraints discussed** so far do not included **general constraints**.
- **Semantic integrity constraints**:
  - Enforces **business rules** within the database (not DDL).
    - EMPLOYEE Salary must not exceed their supervisor
    - Minimum and Maximum hours worked.
    - Employees per project.
- General constraints are implemented using ***Triggers* and *Assertions*** in SQL.

Summary

- Relational Data Model Concepts
- Relational Model Constraints
- DB Operations and Referential Integrity

Questions?

Next Lecture

- SQL Lecture One (data definitions).

Reading

- Chapter 3 from ***Fundamentals of Database Systems***
- Chapter 6 from ***Fundamentals of Database Systems*** for next lecture.