

Relational Data Model

- a data model in which all data is modelled as relations (tables)
 - a way of looking at data
- a prescription for a way of
 - representing data
 - manipulating data (relational algebra)
 - representing integrity constraints

CS5225

1

A Relation

A relation contains a SET of tuples

PARTS(Name: String; Price: Real; Category: String; Manufacturer: String)

Attribute names			
Name	Price (\$)	Category	Manufacturer
Gizmo	19.99	gadgets	GizmoWorks
Power gizmo	29.99	gadgets	GizmoWorks
SingleTouch	149.99	photography	Canon
MultiTouch	203.99	household	Hitachi

Tuples (record)

Each attribute has an atomic type

CS5225

2

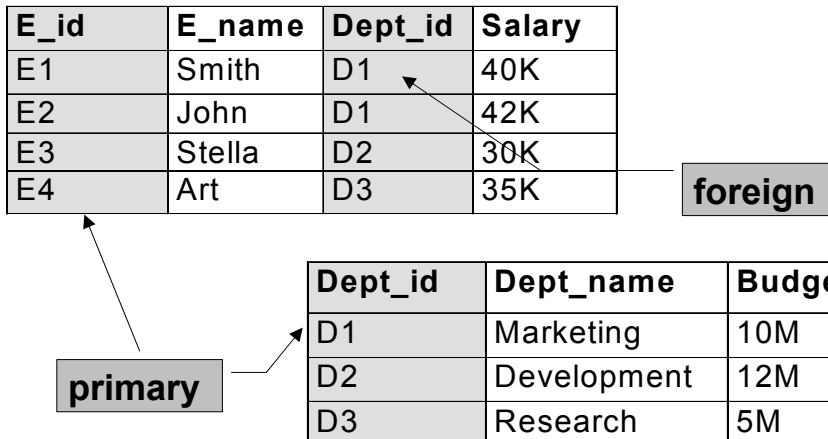
Integrity

- restrictions on data defined by users
 - on individual tables
 - $\text{age} > 18; \text{salary} < 100\text{k}$
 - on more than one table
 - *if* budget < 10M *then* salary < 50k
- implicit in the data model

Integrity Constraints (ICs)

- IC: condition that *must be* true for *any* instance of the database
 - e.g., *domain constraints*
 - Each attribute has values taken from a *domain*. ICs are specified when schema is defined.

Primary and foreign keys



CS5225

5

More Integrity Constraints

- *Key constraints*: each tuple must be distinct. A key is a subset of fields that uniquely identifies a tuple (*superkey*), and for which no subset of the key has this property.
- *Referential integrity constraints*: a field in one relation may refer to a tuple in another relation by including its key (*foreign key*). The referenced tuple must exist in the other relation for the database instance to be valid.
- Typically, a relation may have several *candidate* keys one of which is chosen as the *primary* key.

CS5225

6

Relational Operations

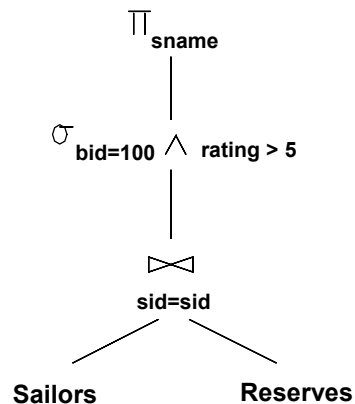
- Selection (σ) Selects a subset of rows from relation.
- Projection (Π) Deletes unwanted columns from relation.
- Join (\bowtie) Allows us to combine two relations.
- Set-difference ($-$) Tuples in reln. 1, but not in reln. 2.
- Union (\cup) Tuples in reln. 1 and in reln. 2.
- Aggregation (SUM, MIN, etc.) and GROUP BY

CS5225

7

Example

```
SELECT S.sname
FROM Reserves R, Sailors S
WHERE R.sid=S.sid AND
      R.bid=100 AND S.rating>5
```



CS5225

Equality Joins With One Join Column

```
SELECT *  
FROM Reserves R, Sailors S  
WHERE R.sid=S.sid
```

- In algebra: $R \bowtie S$.
- Most frequently used operation; very costly operation.

Join Example

<u>sid</u>	<u>sname</u>	<u>rating</u>	<u>age</u>
22	dustin	7	45.0
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

<u>sid</u>	<u>bid</u>	<u>day</u>	<u>rname</u>
31	101	10/11/96	lubber
58	103	11/12/96	dustin

Join Example

<u>sid</u>	sname	rating	age
22	dustin	7	45.0
28	yuppy	9	35.0
31	lubber	8	55.5
44	guppy	5	35.0
58	rusty	10	35.0

<u>sid</u>	<u>bid</u>	<u>day</u>	rname
31	101	10/11/96	lubber
58	103	11/12/96	dustin

<u>sid</u>	<u>sname</u>	<u>rating</u>	<u>age</u>	<u>bid</u>	<u>day</u>	<u>rname</u>
31	lubber	8	55.5	101	10/11/96	lubber
58	rusty	10	35.0	103	11/12/96	dustin

Simple Nested Loops Join

- For each tuple in the outer relation R, we scan the entire inner relation S.

```
foreach tuple r in R do
  foreach tuple s in S do
    if r.sid == s.sid then add <r, s> to result
```