# Grouping, E/R, and updates

CSE 444 section, July 1, 2010

## Today

- Practice with grouping and aggregation
- Database design with E/R diagrams
- Modifying the database

# Document index database

**Author** (aid, name)

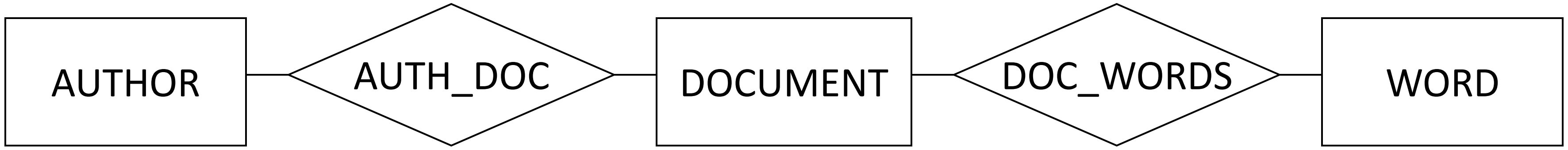
**Auth_Doc** (aid, did)

**Document** (did, title)

**Doc_Word** (did, word)

**Word** (word)

Underlined = key (unique identifier for a tuple)



## Find authors who wrote ≥ 20 docs

# Find authors who wrote ≥ 20 docs

This could work:

**SELECT** name

**FROM** Author a

**WHERE** 20 <= (**SELECT** COUNT(*) **FROM** Auth_Doc ad **WHERE** ad.aid = a.aid)

## Find authors who wrote ≥ 20 docs

Use grouping to eliminate the subquery:

```
SELECT name
FROM Author a, Auth_Doc ad
WHERE a.aid = ad.aid
GROUP BY a.aid, a.name
HAVING COUNT(*) >= 20
```

# Find authors who wrote ≥ 20 docs

Use grouping to eliminate the subquery:

**SELECT** name

**FROM** Author a, Auth_Doc ad

**WHERE** a.aid = ad.aid

**GROUP BY** a.aid, a.name $\longleftarrow$ One row per (a.aid, a.name) pair

**HAVING** COUNT(*) >= 20

# Find authors who wrote ≥ 20 docs

Use grouping to eliminate the subquery:

```
SELECT name
FROM Author a, Auth_Doc ad
WHERE a.aid = ad.aid
GROUP BY a.aid, a.name
HAVING COUNT(*) >= 20  ⟵  Only groups that
                          combine ≥ 20
                          tuples will match
```

# Find authors who wrote ≥ 20 docs

Use grouping to eliminate the subquery:

**SELECT** name

**FROM** Author a, Auth_Doc ad

**WHERE** a.aid = ad.aid

**GROUP BY** a.aid, a.name ⟵ If aid is the key, why group by name?

**HAVING** COUNT(*) >= 20

## If we deleted a.name…

ERROR: Column 'name' is invalid in the select list because it is not contained in either an aggregate function or the GROUP BY clause.

## Finding literate authors

How can we find authors who use more than 10,000 distinct words?

## Authors who use > 10,000 words

```
SELECT name
FROM Author a, Auth_Doc ad,
          Doc_Words dw
WHERE a.aid = ad.aid AND ad.did = dw.did
GROUP BY a.aid, a.name
HAVING COUNT(DISTINCT word) > 10000
```

# Authors who use > 10,000 words

```
SELECT name
FROM Author a, Auth_Doc ad,
        Doc_Words dw
WHERE a.aid = ad.aid AND ad.did = dw.did
GROUP BY a.aid, a.name
HAVING COUNT(DISTINCT word) > 10000
```

→ What does DISTINCT mean within COUNT?

## More examples

- For each author, give the total number of words in all documents he has (co-)written.
- For each author, give the average length in words of his documents.
- Give the author with the longest average documents.

## Total word count by author

```
SELECT name, COUNT(word)
FROM Author a, Auth_Doc ad, Doc_Word dw
WHERE a.aid = ad.aid AND ad.did = dw.did
GROUP BY a.aid, a.name
```

## Average word count by author

```
SELECT name, COUNT(word) / COUNT(DISTINCT
   did) AS avg_count
FROM Author a, Auth_Doc ad, Doc_Word dw
WHERE a.aid = ad.aid AND ad.did = dw.did
GROUP BY a.aid, a.name
```

## Wordiest-on-average author

```
SELECT TOP 1 name, COUNT(word) /
   COUNT(DISTINCT did) AS avg_count
FROM Author a, Auth_Doc ad, Doc_Word dw
WHERE a.aid = ad.aid AND ad.did = dw.did
GROUP BY a.aid, a.name
ORDER BY avg_count DESC
```

## Try these at home

- All words used by at least 10 authors
- The most frequently used word
- The longest document
- Authors of the longest document

## Today

- Practice with grouping and aggregation
- Database design with E/R diagrams
- Modifying the database

## Why use E/R diagrams?

# E/R basics

- Concepts and symbols
  - Entity vs. entity set
  - Attributes
  - Relationship
  - Arrows
- ISA
  - Difference from OOP in C++/Java

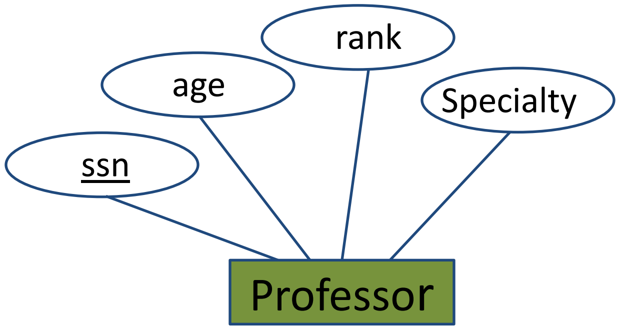
## From English to E/R diagrams

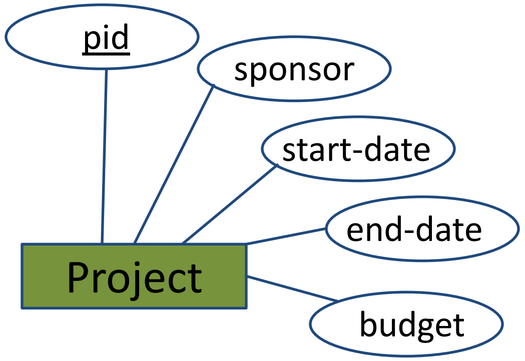
- Each project is managed by one professor (principal investigator)
- A professor can manage multiple projects

Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

## From English to E/R diagrams

- Each project is managed by one professor (principal investigator)
- A professor can manage multiple projects

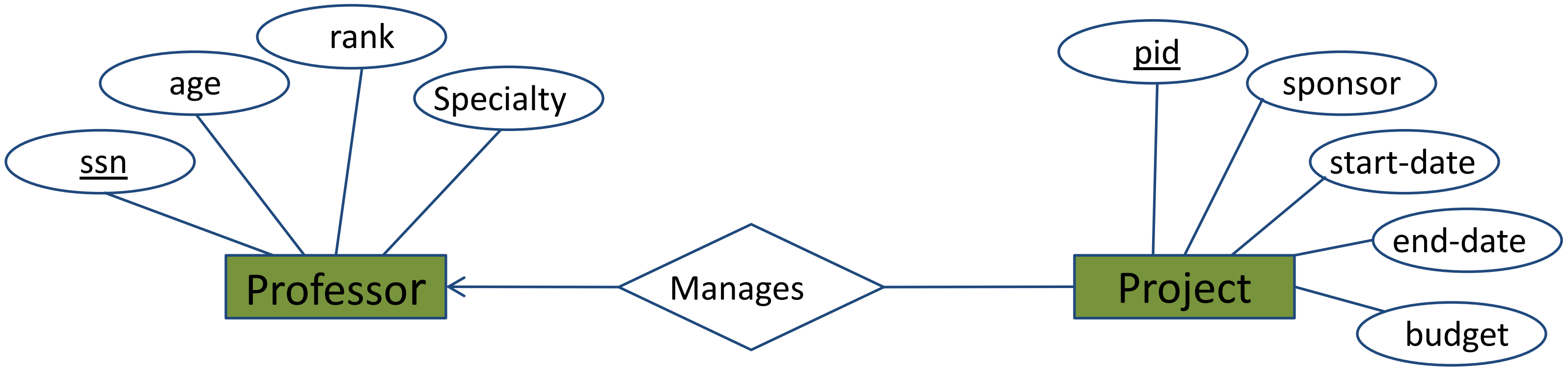




Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

## From English to E/R diagrams

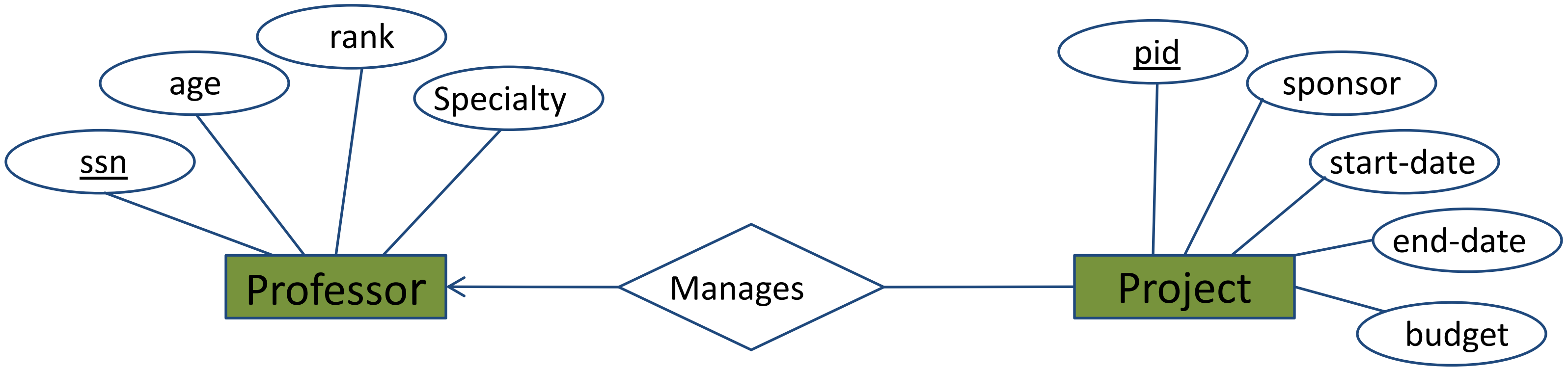
- Each project is managed by one professor (principal investigator)
- A professor can manage multiple projects



Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

# From English to E/R diagrams

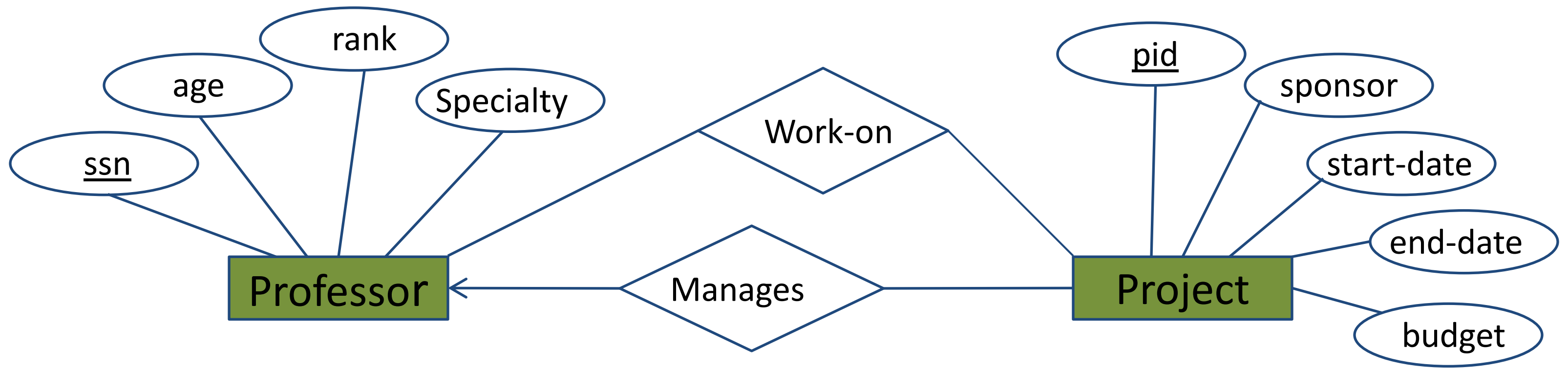
- Each project is **worked on** by one or more professors
- Professors can **work on** multiple projects



Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

# From English to E/R diagrams

- Each project is **worked on** by one or more professors
- Professors can **work on** multiple projects



Example courtesy: Database Management Systems, 3rd E, R. Ramakrishnan and J. Gehrke

## Today

- Practice with grouping and aggregation
- Database design with E/R diagrams
- Modifying the database

## Modifying the database

Three kinds of modifications in SQL:

- insertions
- updates
- deletions

Sometimes they are all called “updates”

# Insertions

General form:

```
INSERT   INTO   R(A1,…., An)   VALUES  (v1,…., vn)
```

# Insertions

**Product** (<u>name</u>, listPrice, category)
**Purchase** (buyer, seller, product, price)

Example: Insert a new purchase to the database:

```
INSERT  INTO  Purchase (buyer, seller, product, price)
        VALUES  ('Joe', 'Fred', 'wakeup-clock-espresso-machine',
                 199.99)
```

Missing attributes → NULL.
May drop attribute names if you give them in order.

## Inserting results of a query

```
INSERT   INTO   Product (name)

    SELECT  DISTINCT  Purchase.product
    FROM      Purchase
    WHERE   Purchase.date > "10/26/01";
```

The query replaces the VALUES keyword.
Here we insert *many* tuples into Product

# Updates

Example:

```
UPDATE   Product
SET    price = price/2
WHERE  Product.name  IN
                (SELECT product
                 FROM    Purchase
                 WHERE  Date ='Oct, 25, 1999');
```

WHERE works the same as in a query (SELECT).
It chooses the tuples whose values are to be updated

# Deletions

Similar to UPDATE but without the SET clause:

```
DELETE   FROM   Purchase

WHERE    seller = 'Joe'   AND
         product = 'Brooklyn Bridge'
```

Always specify a WHERE clause (in fact, write it first!)
Otherwise, *every tuple* will be deleted!