INFO20003 Database Systems

Dr Renata Borovica-Gajic

Lecture 04

Relational Model &

Translating ER diagrams

Week 2

- Relational Model

- Keys & Integrity Constraints

- Translating ER to Logical and Physical Model

Readings: Chapter 3, Ramakrishnan & Gehrke, Database Systems

- **Data Model** allows us to translate real world things into structures that a computer can store

- Many models: <u>Relational</u>, ER, O-O, Network, Hierarchical, etc.

- **Relational Model:**

  – Rows & Columns (Tuples/records and Attributes/fields)

  – Keys & Foreign Keys to link Relations

Enrolled

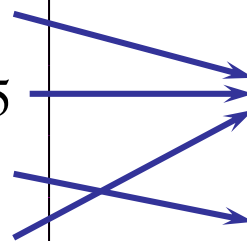| sid	cid	grade
53666	Carnatic101	5
53666	Reggae203	5.5
53650	Topology112	6
53666	History105	5

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	5.4
53688	Smith	smith@eecs	18	4.2
53650	Smith	smith@math	19	4.8

- **_Relational database_**_: a set of relations._

- **_Relation_**_: made up of 2 parts:_
  - **_Schema_** : specifies name of relation, plus name and type of each column (attribute).

    Example: Students(_sid_: string, _name_: string, _login_: string, _age_: integer, _gpa_: real)
  - **_Instance_** : a **table,** with rows and columns.

    #rows = _cardinality_

    #fields = _degree (or arity)_

- You can think of a relation as a _set_ of _rows_ or _tuples_.
  - all rows are _distinct_, no order among rows

Students
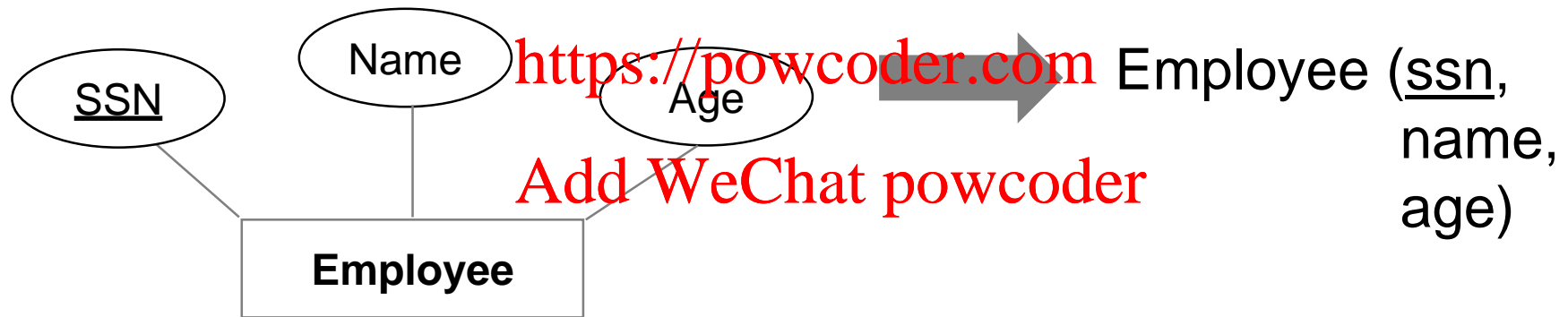
| sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@eecs	18	3.2
53650	Smith	smith@math	19	3.8

Cardinality = 3, degree (arity) = 5, all rows distinct

In logical design **entity** set becomes a **relation.** Attributes become attributes of the relation.

Conceptual Design:

Name

SSN

Age

Employee

Logical Design:

Employee (ssn, name, age)

In physical design we choose data types

1. Conceptual Design:
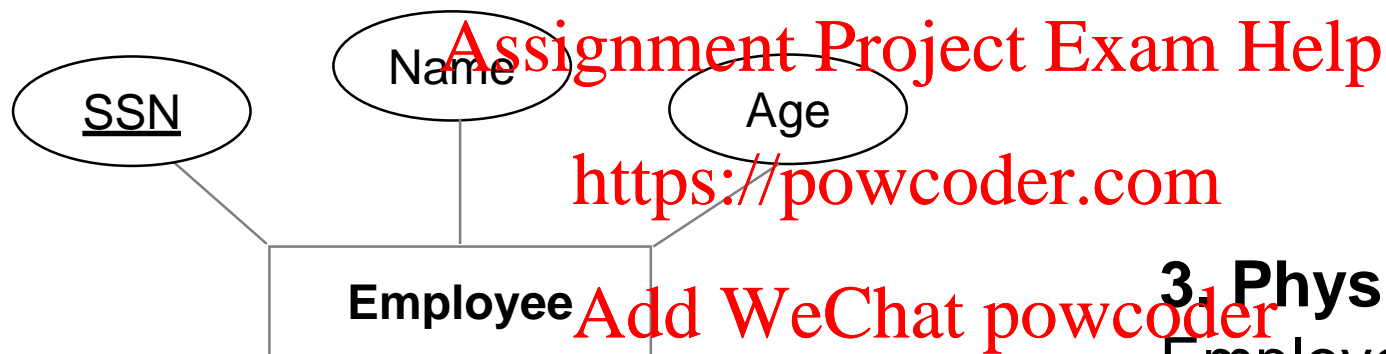
SSN

Name

Age

Employee

2. Logical Design:

Employee (ssn, name, age)

3. Physical Design:
Employee
(ssn CHAR(11),
name VARCHAR(20),
age INTEGER)

1. Conceptual Design

Assignment Project Exam Help

2. Logical Design

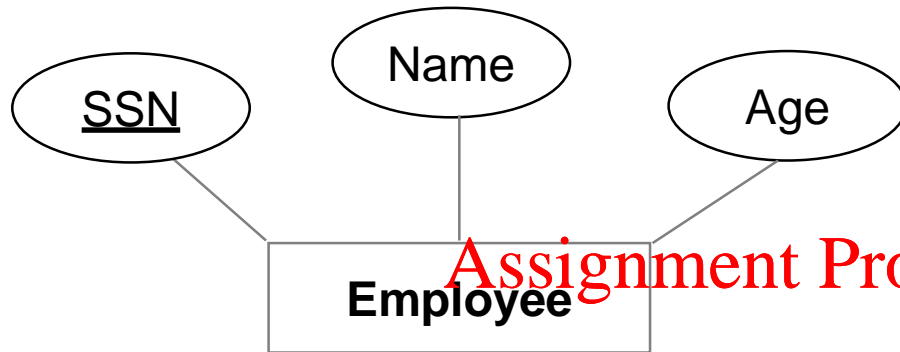https://powcoder.com

3. Physical Design

Add WeChat powcoder

4. Implementation

5. Create Instance

1. Conceptual Design:

4. Implementation:

CREATE TABLE Employee
           (ssn CHAR(11),
           name VARCHAR(20),
           age INTEGER,
           PRIMARY KEY (ssn))

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

2. Logical Design:

Employee (ssn,
           name,
           age)

3. Physical Design:

Employee
(ssn CHAR(11),
 name VARCHAR(20),
 age INTEGER)

5. Instance:

EMPLOYEE

ssn	name	age
0983763423	John	30
9384392483	Jane	30
3743923483	Jill	20

Example: Creating the Students relation.

CREATE TABLE Students
(sid CHAR(20),
name CHAR(20),
login CHAR(10),
age INTEGER,
gpa FLOAT)

The type (domain) of each field is specified, and enforced by the DBMS whenever tuples are added or modified.

- Relational Model & SQL overview

- **Keys & Integrity Constraints**

- Translating ER to Logical and Physical Model

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Readings: Chapter 3, Ramakrishnan & Gehrke, Database Systems

- Keys are a way to associate tuples in different relations

- Keys are one form of **integrity constraint (IC)**

- **Example:** *Only students can be enrolled in subjects.*

Enrolled

sid	cid	grade
53666	15-101	C
53666	18-203	B
53650	15-112	A
53666	15-105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@cs	18	3.2
53650	Smith	smith@math	19	3.8

FOREIGN Key

PRIMARY Key

© University of Melbourne

- A set of fields is a ***superkey*** if no two distinct tuples can have same values in all key fields

- A set of fields is a ***key*** for a relation if it is a superkey and no subset of the fields is a superkey (minimal subset)

- Out of all keys *one* is chosen to be the ***primary key*** of the relation. Other keys are called ***candidate*** keys

- Each relation has a primary key

- **Your turn:**

  1. *Is sid* a key for Students?

  2. What about *name*?

  3. Is the set {*sid, gpa*} a superkey? Is the set {*sid, gpa*} a key?

  4. Find a primary key from this set {sid, login}

- There are possibly many _candidate keys_ (specified using UNIQUE), one of which is chosen as the *primary key*. Keys must be chosen carefully.

Example:

For a given student and course, there is a single grade.

```
CREATE TABLE Enrolled
 (sid CHAR(20)
  cid  CHAR(20),
  grade CHAR(2),
  PRIMARY KEY (sid,cid))
```

VS.

```
CREATE TABLE Enrolled
 (sid CHAR(20)
  cid  CHAR(20),
  grade CHAR(2),
  PRIMARY KEY (sid),
  UNIQUE (cid, grade))
```

"Students can take only one course, and no two students in a course receive the same grade."

- ***Foreign key*** : A set of fields in one relation that is used to 'refer' to a tuple in another relation. Foreign key must correspond to the primary key of the other relation.

- If all foreign key constraints are enforced in a DBMS, we say ***referential integrity*** is achieved.

Example: *Only students listed in the Students relation should be allowed to enroll in courses.*

- *sid* is a foreign key referring to Students

```
CREATE TABLE Enrolled
(sid CHAR(20),
 cid CHAR(20),
 grade CHAR(2),
 PRIMARY KEY (sid,cid),
 FOREIGN KEY (sid) REFERENCES Students )
```

Enrolled

sid	cid	grade
53666	15-101	C
53666	18-203	B
53650	15-112	A
53666	15-105	B

Students

sid	name	login	age	gpa
53666	Jones	jones@cs	18	3.4
53688	Smith	smith@cs	18	3.2
53650	Smith	smith@math	19	3.8

- Consider Students and Enrolled; *sid* in Enrolled is a foreign key that references Students.

- What should be done if an Enrolled tuple with a non-existent student id is inserted? (*Reject it!*)

- What should be done if a Students tuple is deleted?

  - Also delete all Enrolled tuples that refer to it?

  - Disallow deletion of a Students tuple that is referred to?

  - Set sid in Enrolled tuples that refer to it to a *default sid*?

  - (In SQL, also: Set sid in Enrolled tuples that refer to it to a special value *null,* denoting `*unknown*´ or `*inapplicable*´.)

- Note: Similar issues arise if primary key of Students tuple is updated.

- **IC**: condition that must be true for *any* instance of the database; e.g., *domain constraints*.
  - ICs are specified when schema is defined.
  - ICs are checked when relations are modified.

- A *legal* instance of a relation is one that satisfies all specified ICs.
  - DBMS should not allow illegal instances.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- Relational Model & SQL overview

- Keys & Integrity Constraints

- **Translating ER to Logical and Physical Model**
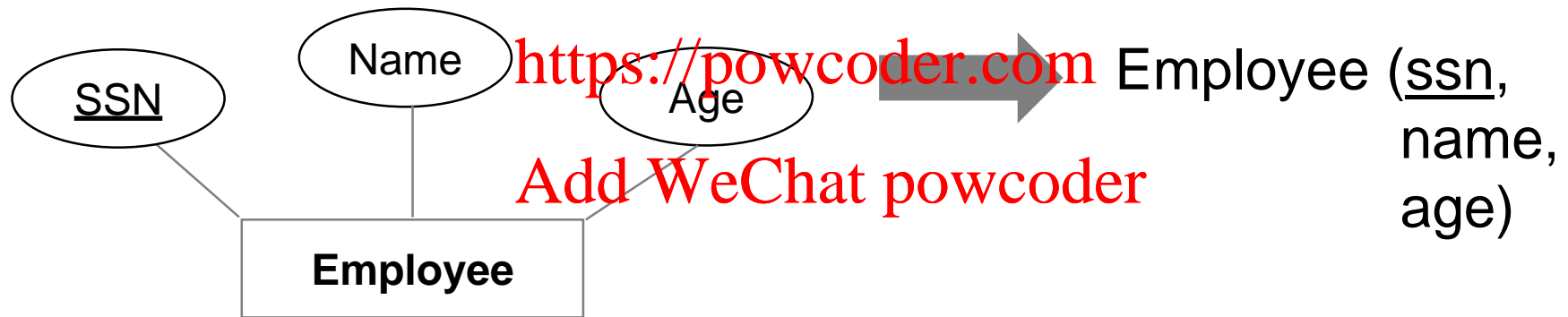
Readings: Chapter 3, Ramakrishnan & Gehrke, Database Systems

In logical design **entity** set becomes a **relation.** Attributes become attributes of the relation.

Conceptual Design: **Logical Design**:

SSN
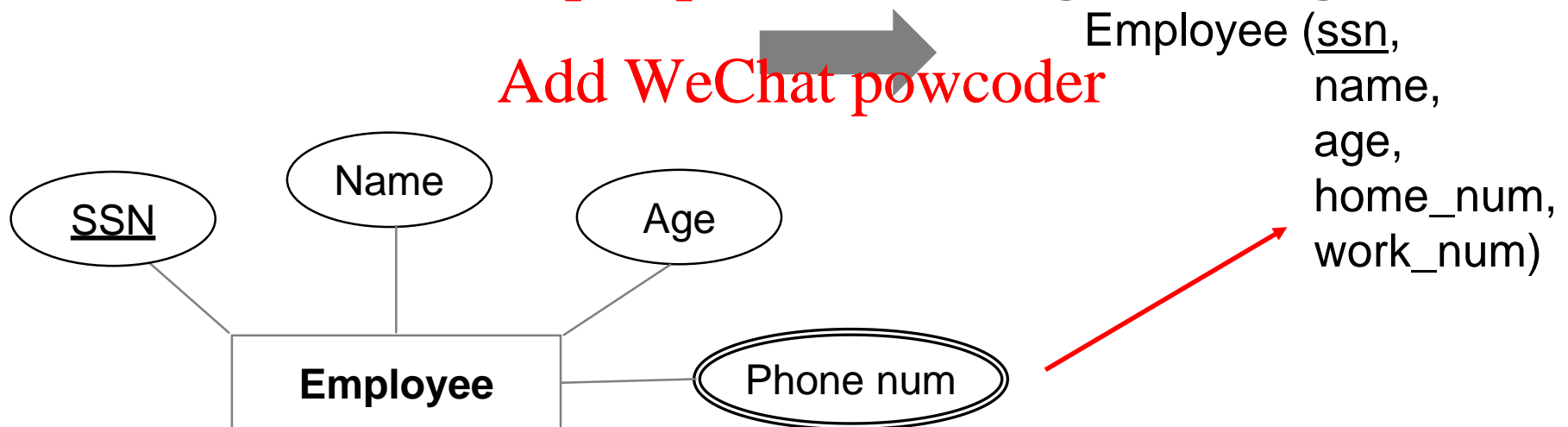
Name

Age

Employee

Employee (<u>ssn</u>, name, age)

- Multi-valued attributes need to be unpacked (flattened) when converting to logical design. *There is an alternative of creating a lookup table discussed in the next lecture.*

Example:

For employees we need to capture their home phone number and work phone number.

Assignment Project Exam Help

Conceptual Design: https://powcoder.com **Logical Design**:

Add WeChat powcoder

Employee (<u>ssn</u>,
name,
age,
home_num,
work_num)

SSN
Name
Age
Employee
Phone num

Multi-valued attribute

THE UNIVERSITY OF
MELBOURNE

- Composite attributes need to be unpacked (flattened) when converting to logical design.
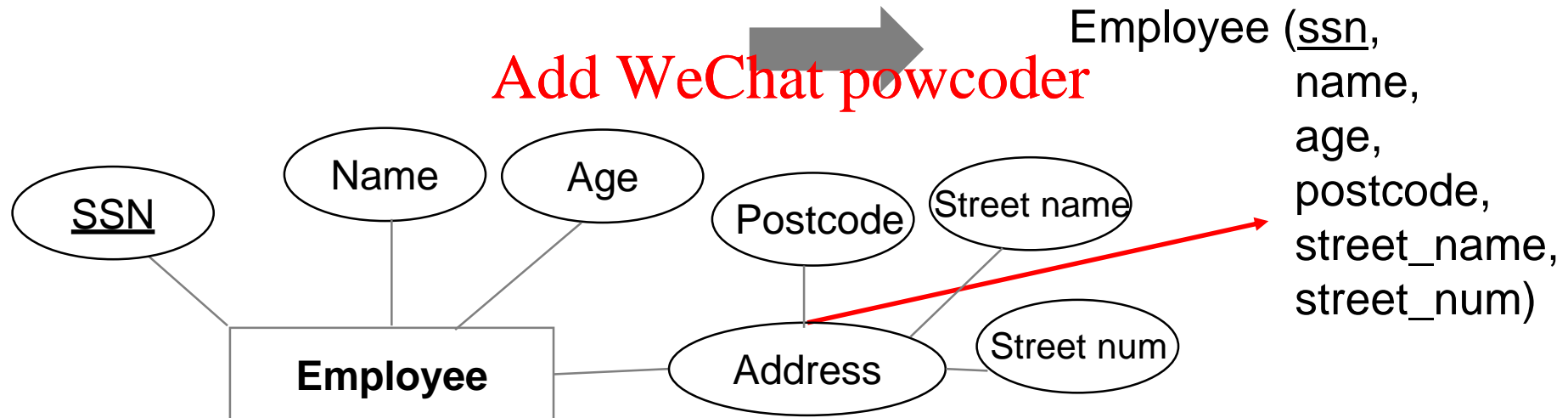
Example:

For employees we need to capture an address consisting of a postcode, street name and number.
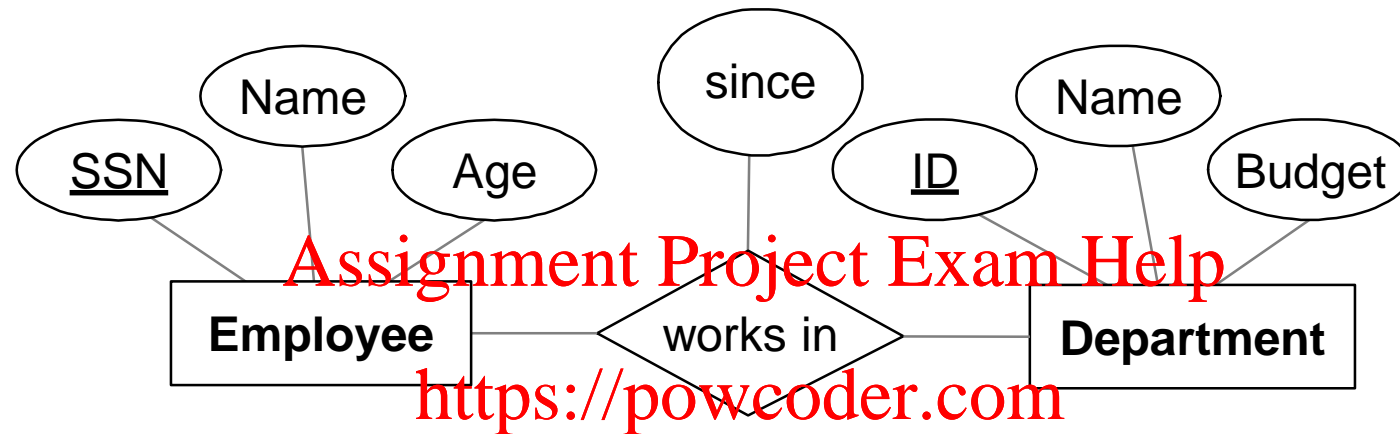
Assignment Project Exam Help

Conceptual Design: https://powcoder.com **Logical Design**:

Add WeChat powcoder

Employee (<u>ssn</u>, name, age, postcode, street_name, street_num)

SSN

Name   Age

Postcode   Street name

Street num

Employee   Address

Composite attribute

Conceptual Design:

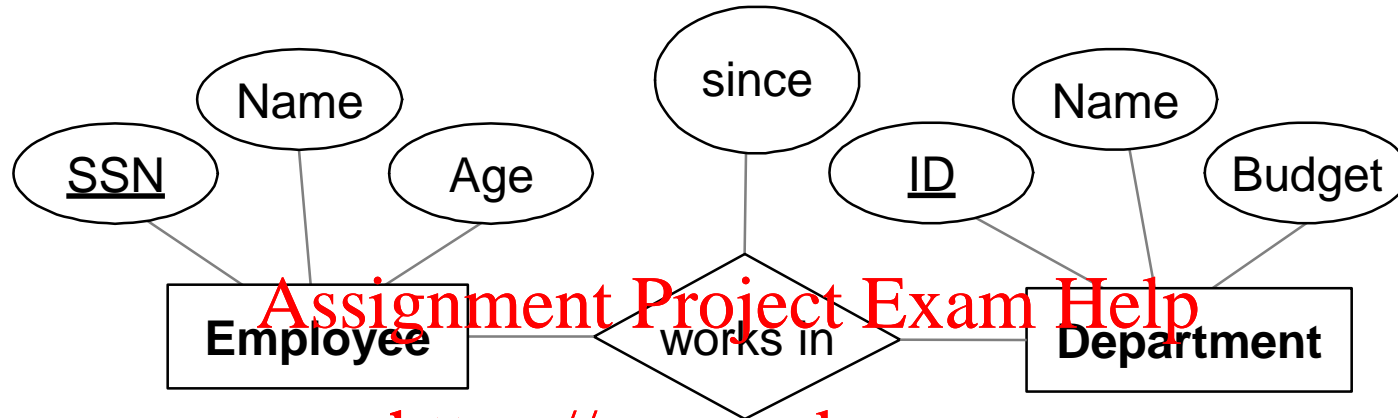
Logical Design:

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

In translating a **many-to-many** relationship set to a relation, attributes of a *new* relation must include:

1. Keys for each participating entity set (as foreign keys). This set of attributes forms a *superkey* of the relation.
2. All descriptive attributes.

Conceptual Design:

Logical Design:

Employee (ssn, name age)

Department (did, dname, budget)

Add WeChat powcoder

Works_In (**ssn**, **did**, since)

This is called an associative entity

Keys from connecting entities become PFK

Note: Underline = PK, italic and underline = FK, underline and bold = PFK

Logical Design:

Employee (<u>ssn</u>, name, age)

Department (<u>did</u>, dname, budget)

Works_In (**<u>ssn</u>**, **<u>did</u>**, since)

Note: Underline = PK,
italic and underline = FK,
underline and bold = PFK

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

Physical Design:

Employee
(<u>ssn</u> CHAR(11),
name VARCHAR(20),
age INTEGER)

Department
(<u>did</u> INTEGER,
dname VARCHAR(20),
budget FLOAT)

Works_In(
<u>ssn</u> CHAR(11),
<u>did</u> INTEGER,
since DATE)

Logical Design:

Employee (<u>ssn</u>, name, age)

Department (<u>did</u>, dname, budget)

Works_In (**<u>ssn</u>**, **<u>did</u>**, since)

Note: Underline = PK,
italic and underline = FK,
underline and bold = PFK

Implementation:

```
CREATE TABLE Employee
        (ssn CHAR(11),
         name VARCHAR(20),
         age INTEGER,
         PRIMARY KEY  (ssn))
```

```
CREATE TABLE Department
        (did INTEGER,
         dname VARCHAR(20),
         budget FLOAT,
         PRIMARY KEY  (did))
```

```
CREATE TABLE Works_In
     (ssn  CHAR(11),
      did  INTEGER,
      since  DATE,
      PRIMARY KEY (ssn, did),
      FOREIGN KEY (ssn) REFERENCES Employee,
      FOREIGN KEY (did) REFERENCES Department)
```

Employee

ssn	name	age
0983763423	John	30
9384392483	Jane	30
3743923483	Jill	20

Department

did	dname	budget
101	Sales	10K
105	Purchasing	20K
108	Databases	1000K

Works_In

ssn	did	since
0983763423	101	1 Jan 2003
0983763423	108	2 Jan 2003
9384392483	108	1 Jun 2002

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

In translating a many-to-many relationship set to a relation, attributes of the relation must include:

- Keys for each participating entity set (as foreign keys). This set of attributes forms a *superkey* for the relation.
- All descriptive attributes.

Parts

id   name

Suppliers

id   name

Contract

Departments

id   name
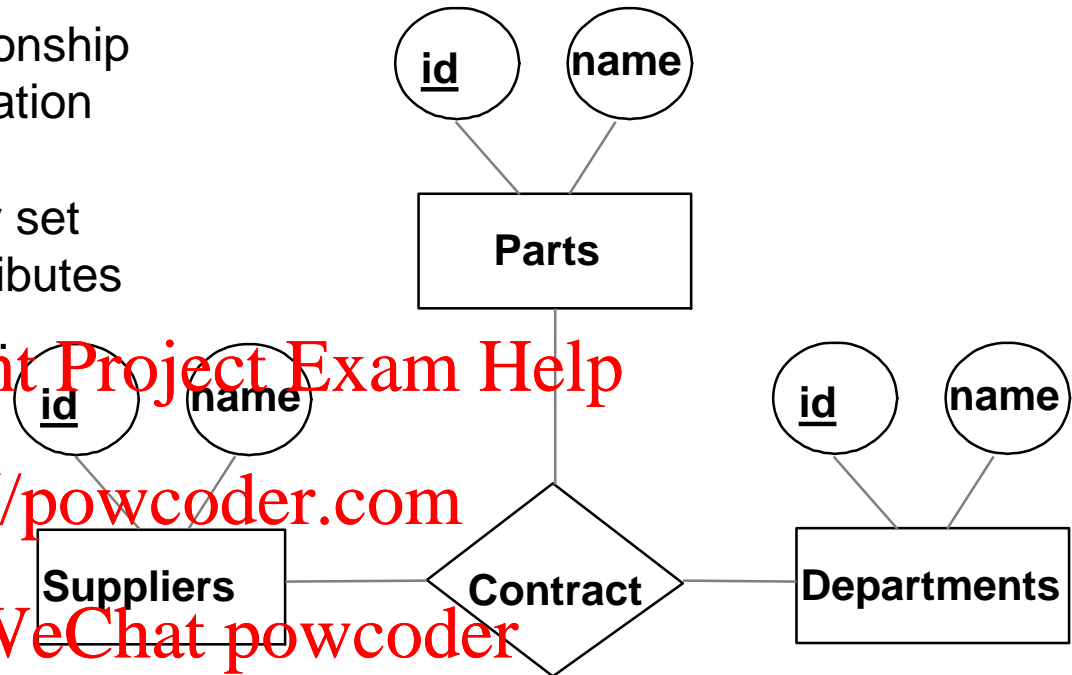
Logical Design:

Contracts (

supplier_id,

part_id,

department_id)

Note: Underline = PK, italic and underline = FK, underline and bold = PFK
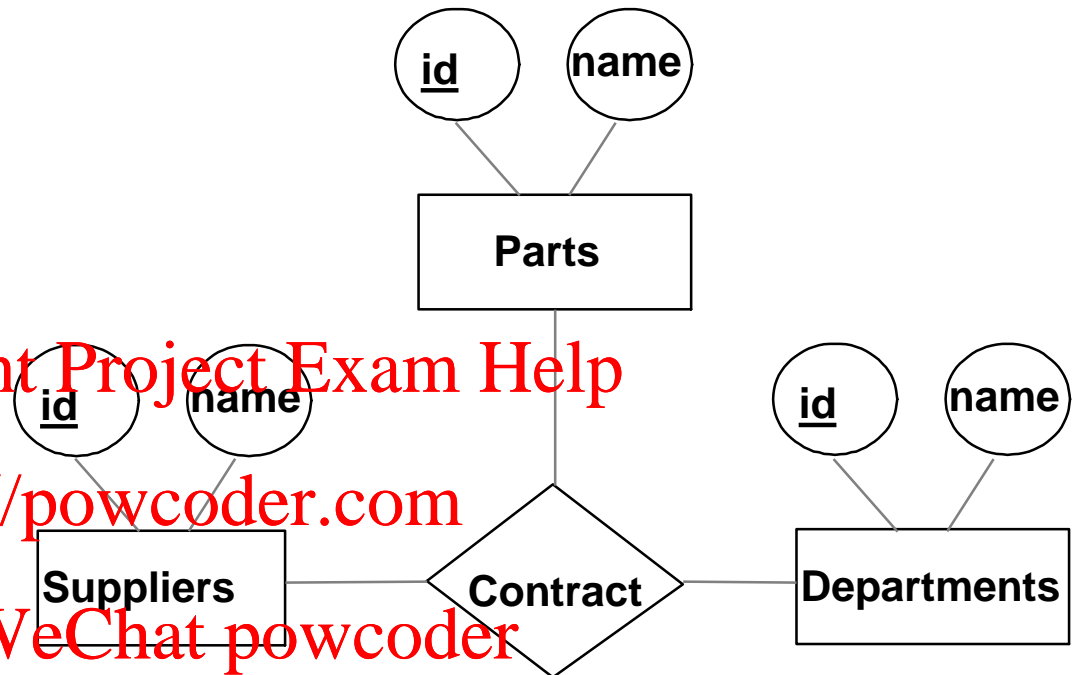
Logical Design:

Contracts (

supplier_id,

part_id,

department_id)

Implementation:

CREATE TABLE Contracts (

supplier_id INTEGER,

part_id INTEGER,

department_id INTEGER,

PRIMARY KEY (supplier_id, part_id, department_id),

FOREIGN KEY (supplier_id) REFERENCES Suppliers,

FOREIGN KEY (part_id) REFERENCES Parts,

FOREIGN KEY (department_id) REFERENCES Departments)

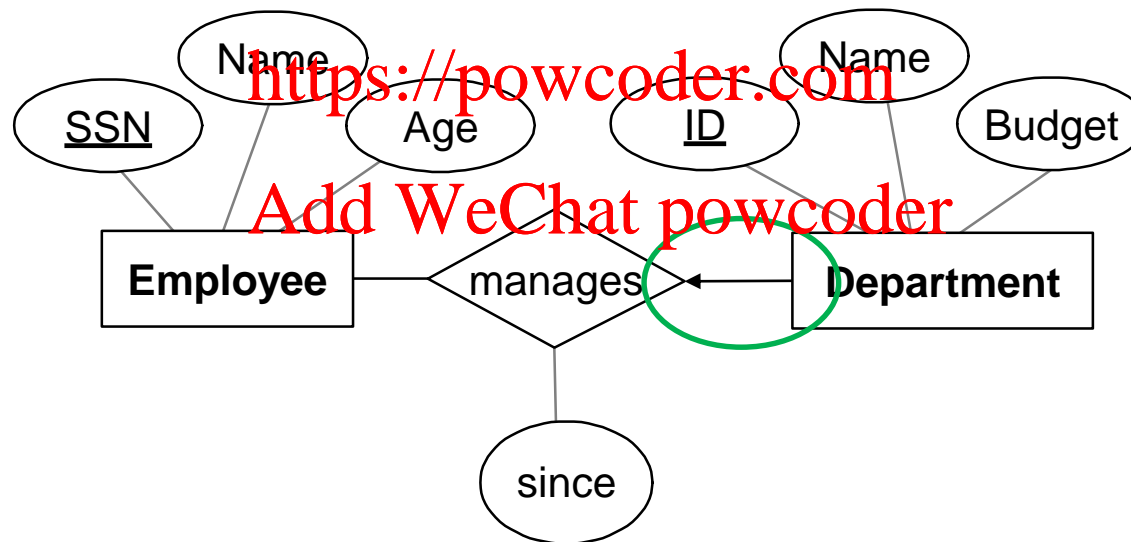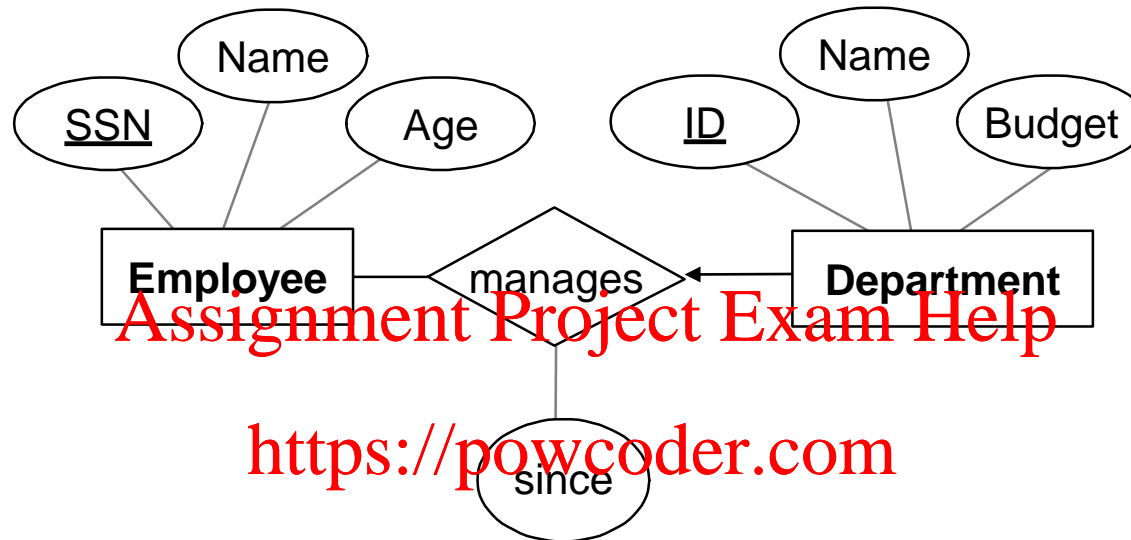- Each department has at most one manager, according to the *key constraint* on Manages.

Assignment Project Exam Help

https://powcoder.com
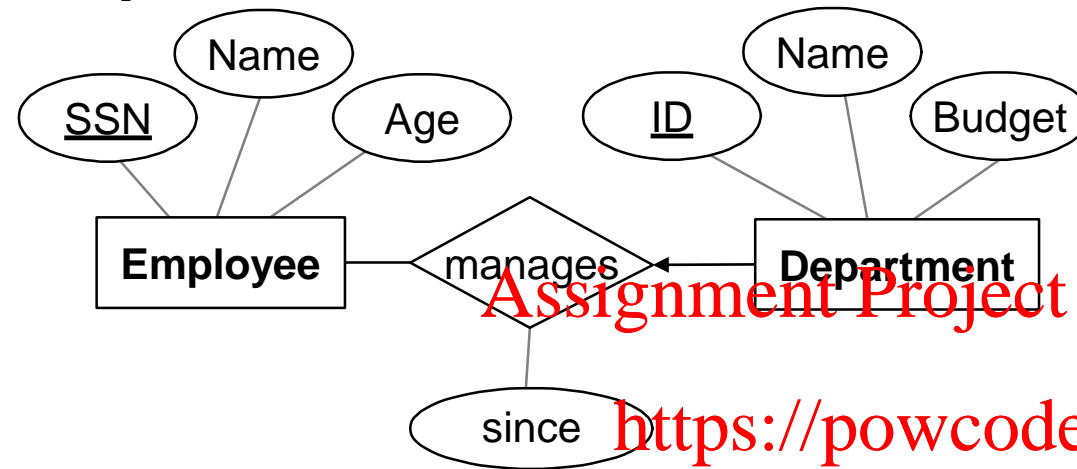
Add WeChat powcoder

Logical Design:

Employee (<u>ssn</u>, name, age)   **VS.**   Employee (<u>ssn</u>, name, age)

Department (<u>did</u>, dname, budget)

Manages (**<u>ssn</u>, <u>did</u>**, since)

Department (<u>did</u>, dname, budget, *<u>ssn</u>*, since)

Note: Underline = PK, italic and underline = FK, underline and bold = PFK

Implementation:

CREATE TABLE Employee
         (ssn CHAR(11),
          name VARCHAR(20),
          age INTEGER,
          PRIMARY KEY (ssn))

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

CREATE TABLE Manages   **vs.**
 (ssn CHAR(11),
 did INTEGER,
 since DATE,
 PRIMARY KEY (ssn, did),
 FOREIGN KEY (ssn)
 REFERENCES Employees,
 FOREIGN KEY (did)
 REFERENCES Departments)

CREATE TABLE Department
          (did INTEGER,
           dname CHAR(20),
           budget FLOAT,
           ssn CHAR(11),
           since DATE,
           PRIMARY KEY (did)
           FOREIGN KEY (ssn)
            REFERENCES Employees)

Which one is better?

- RULE: *Primary key from the many side becomes a foreign key on the one side*
- This is the way to ensure that the key constraint holds

Assignment Project Exam Help

```
CREATE TABLE Department
        (did INTEGER,
         dname CHAR(20),
         budget FLOAT,
         ssn CHAR(11),
         since DATE,
         PRIMARY KEY  (did)
         FOREIGN KEY (ssn)
          REFERENCES Employee)
```
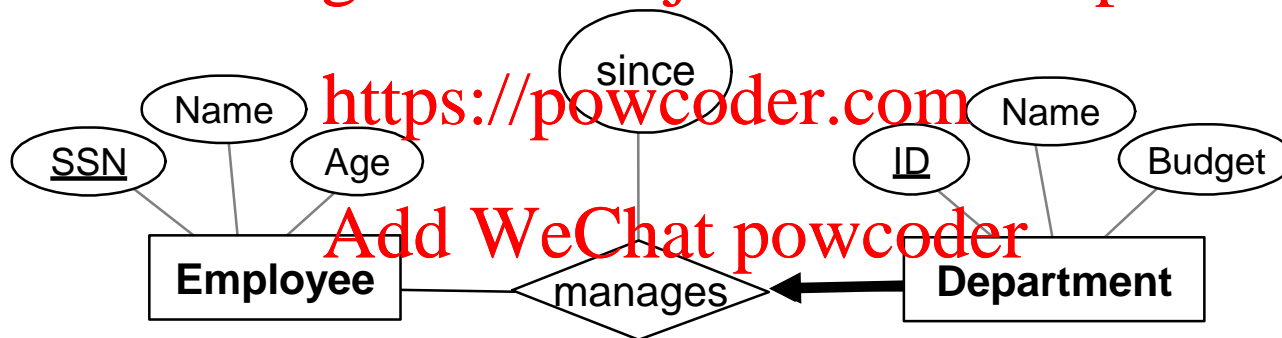
https://powcoder.com

Add WeChat powcoder

Each department will have a *single* manager

- Does every department have a manager?
  - If so, this is a *participation constraint*: the participation of Departments in Manages is said to be *total*.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- We specify total participation with key words NOT NULL
  - **NOT NULL = this field cannot be empty**

```
CREATE TABLE Department (
  did  INTEGER NOT NULL,
  dname  CHAR(20) NOT NULL,
  budget  FLOAT NULL,
  ssn  CHAR(11) NOT NULL,
  since  DATE NULL,
  PRIMARY KEY  (did),
  FOREIGN KEY  (ssn) REFERENCES Employee
    ON DELETE NO ACTION)
```

NOTE: Every time we create a table or draw a physical design we should specify whether attributes are NULL or NOT NULL. We haven't done it in each slide of this lecture due to clarity and lack of space – but don't forget this in your design/implementation!
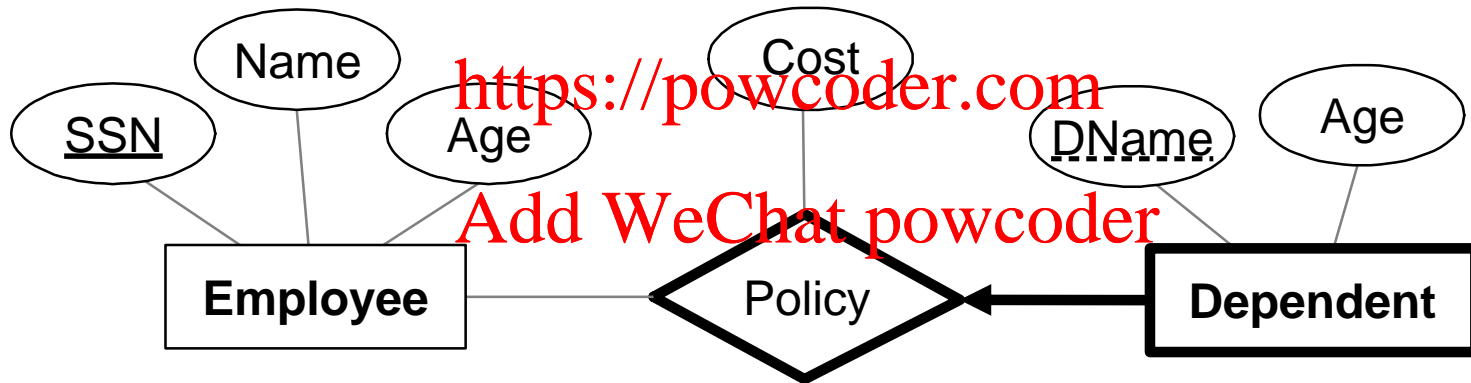
- A **_weak entity_** can be identified uniquely only by considering the primary key of another (*owner*) entity.

- Weak entity set and identifying relationship set are translated into a single table.
  - When the owner entity is deleted, all owned weak entities must also be deleted.

Logical Design:

Dependent (<u>dname</u>, age, cost, **ssn**)

Note: Underline = PK,
italic and underline = FK,
underline and bold = PFK

Implementation:

CREATE TABLE Dependent (
    dname CHAR(20) NOT NULL,
    age INTEGER NULL,
    cost DECIMAL(7,2) NOT NULL,
    ssn CHAR(11) NOT NULL,
    PRIMARY KEY (dname, ssn),
    FOREIGN KEY (ssn) REFERENCES Employees
        ON DELETE CASCADE)

- A tabular representation of data.

- Simple and intuitive, currently the most widely used.

- Integrity constraints can be specified based on application semantics. DBMS checks for violations.

  – Two important ICs: primary and foreign keys

  – In addition, we always have domain constraints.

- Rules to translate ER to logical design (relational model)

- Translate conceptual (ER) into logical & physical design
- Understand integrity constraints
- Use DDL of SQL to create tables with constraints

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

- ER Modelling Example with MySQL Workbench
  - You will need this for workshops/labs (and assessment)

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder