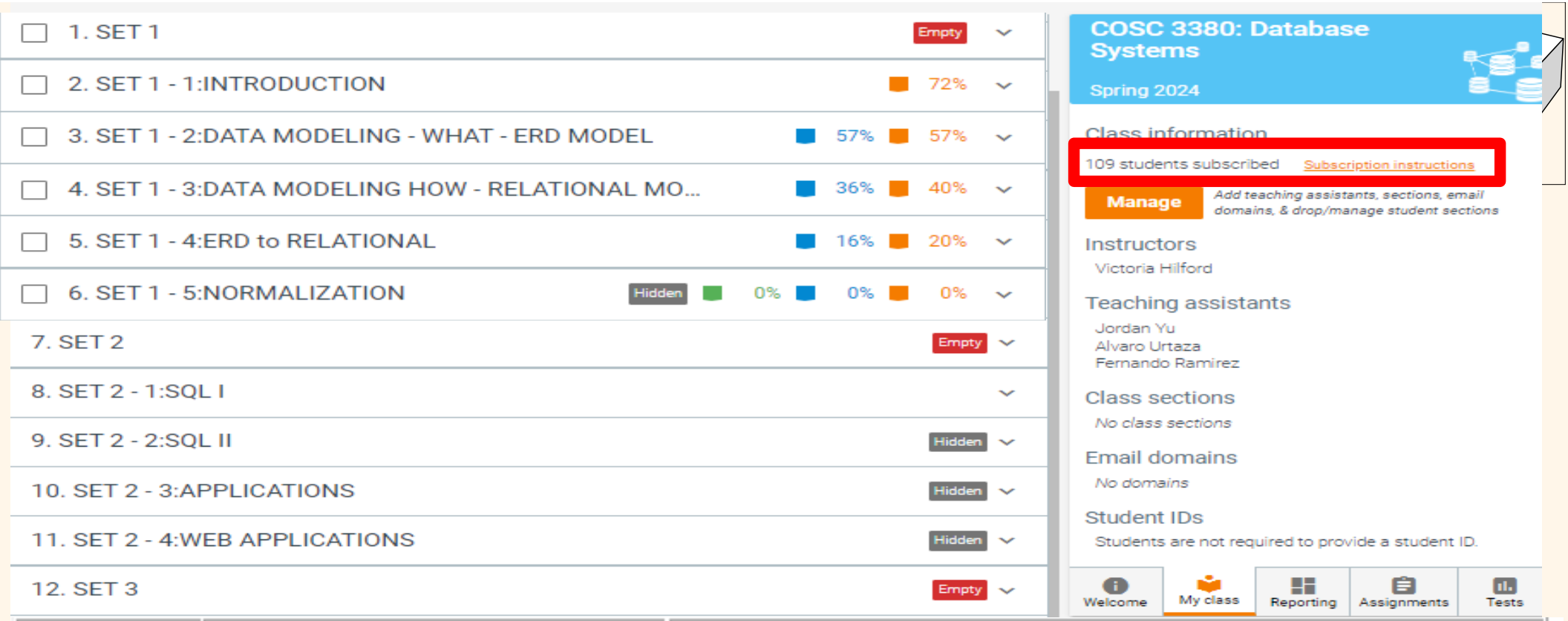
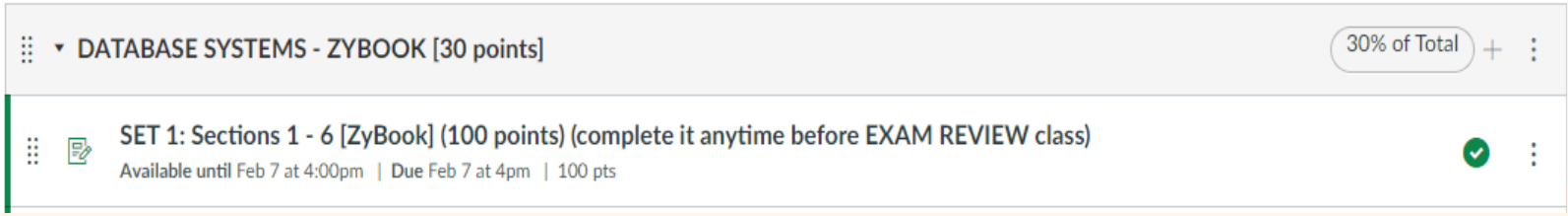


02.07.2024 (7 - We)	TA Download ZyBook SET 1 Sections (4 PM) (PART of 30 points)	EXAM 1 Review (PART of 20 points)



Only 109 out of 120 registered.

UNIVERSITY of HOUSTON

DEPARTMENT OF COMPUTER SCIENCE

COSC 3380 Spring 2024

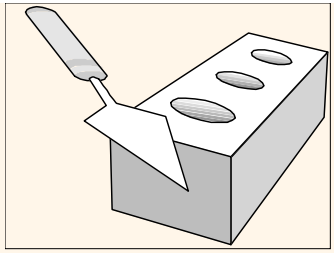
Database Systems

M & W 4:00 to 5:30 PM

Prof. **Victoria Hilford**

PLEASE TURN your webcam ON (must have)

NO CHATTING during LECTURE



COSC 3380

4 to 5:30

PLEASE

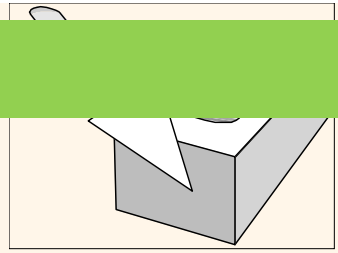
LOG IN

CANVAS

Please close all other windows.

01.29.2024 (4 - Mo)	**ZyBook SET 1 - 4**	**Set 1** LECTURE 4 ERD to RELATIONAL
01.31.2024 (5 - We)	**ZyBook SET 1 - 5**	**Set 1** LECTURE 5 NORMALIZATION
02.05.2024 (6 - Mo)		**EXAM 1 Practice** **(PART of 20 points)**
02.07.2024 (7 - We)	**TA Download** **ZyBook SET 1 Sections** **(4 PM)** **(PART of 30 points)**	**EXAM 1 Review** **(PART of 20 points)**
02.12.2024 (8 - Mo)		**EXAM 1** **(PART of 50 points)**

From 4:00 to 4:07 PM – 7 minutes.



01.29.2024 (4 - Mo)	ZyBook SET 1 - 4	Set 1 LECTURE 4 ERD to RELATIONAL

CLASS PARTICIPATION 20 points — 20% of Total

CLASS 4

Class 4 BEGIN PARTICIPATION
Not available until Jan 29 at 4:00pm | Due Jan 29 at 4:07pm | 100 pts

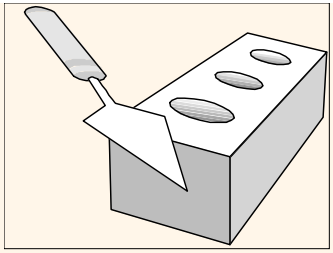
VH, publish

This is a synchronous online class.

Attendance is required.

Recording or distribution of class materials is prohibited.

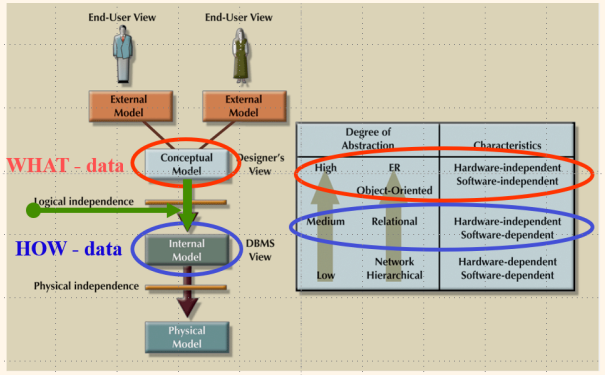
1. At the beginning of selected classes there is an **assessment in the first 10 minutes**. (**beige** BOX in the Detailed Syllabus)
2. At the end of selected classes there is an **assessment in the last 10 minutes**. (**blue** BOX in the Detailed Syllabus)
3. ZyBook sections will be downloaded and used for 30% of Total Score on the dates specified in the Detailed Syllabus.
4. EXAMS are in CANVAS. **No late EXAMS**.
5. I have to be **present in TEAMS in order to take any graded assignment assigned during that class.**



COSC 3380

Class 4

01.29.2024 (4 - Mo)	ZyBook SET 1 - 4	Set 1 LECTURE 4 ERD to RELATIONAL

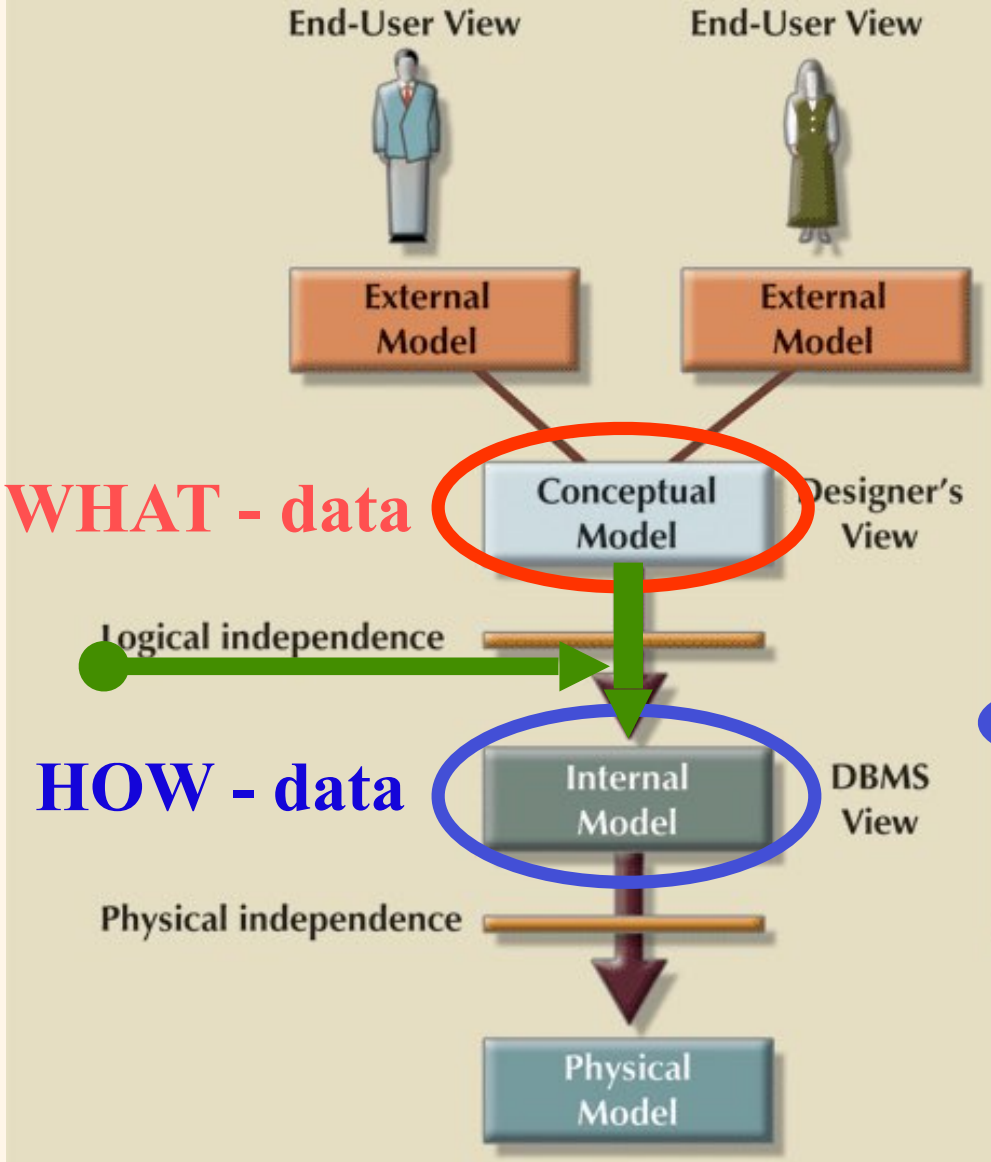


From 4:07 to 5:00 PM – 50 minutes.

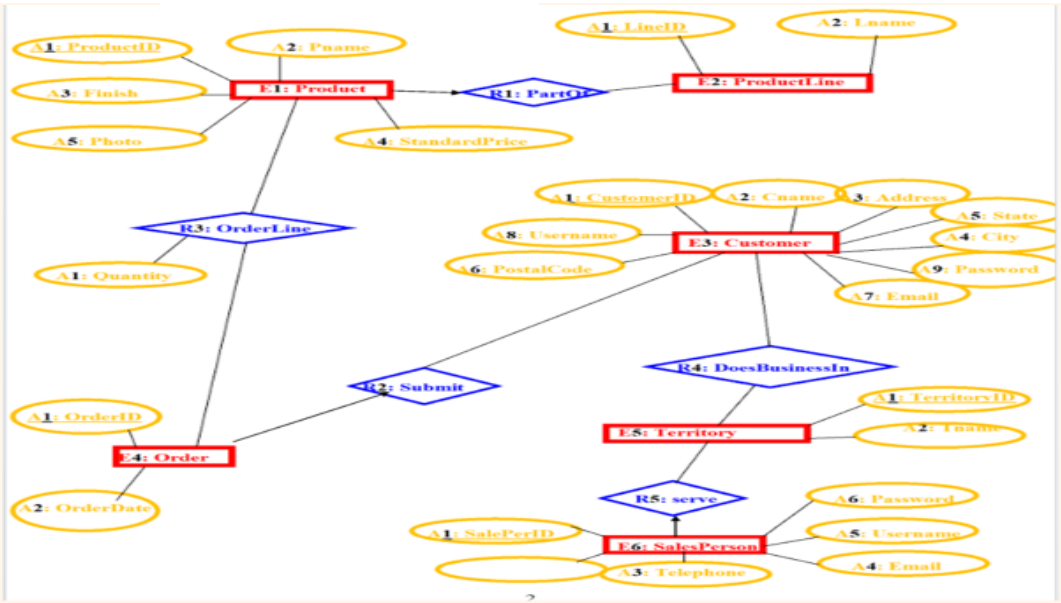
Lecture 4

ER Data Model to Relational Data Model
WHAT to HOW

Data abstraction levels

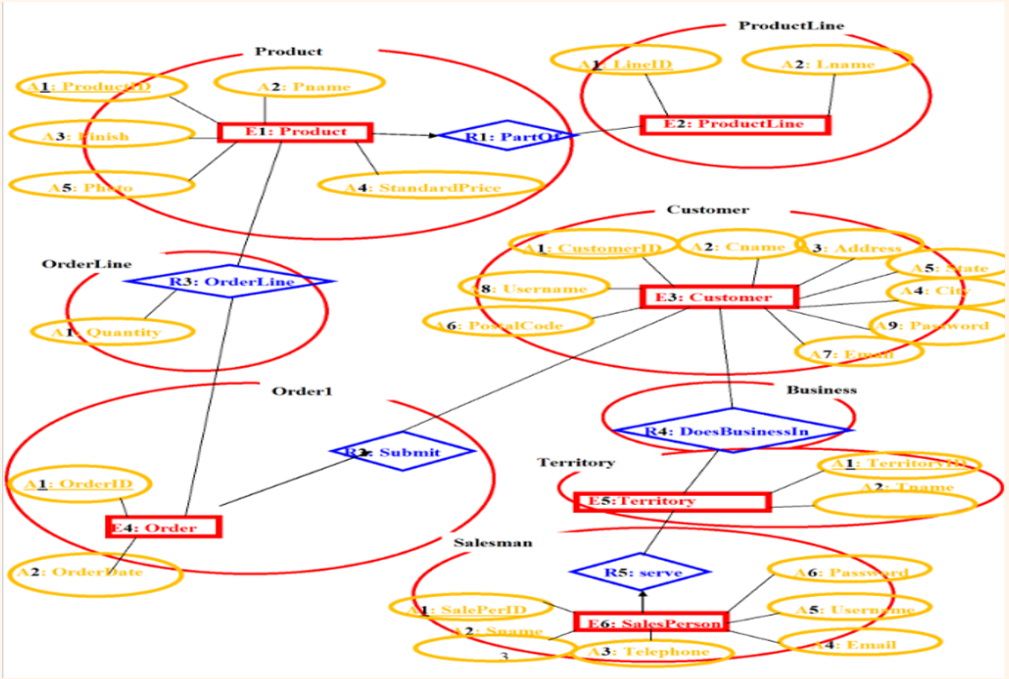


Degree of Abstraction		Characteristics
High	ER Object-Oriented	Hardware-independent Software-independent
Medium	Relational	Hardware-independent Software-dependent
Low	Network Hierarchical	Hardware-dependent Software-dependent



From ER Diagrams *to* Relations

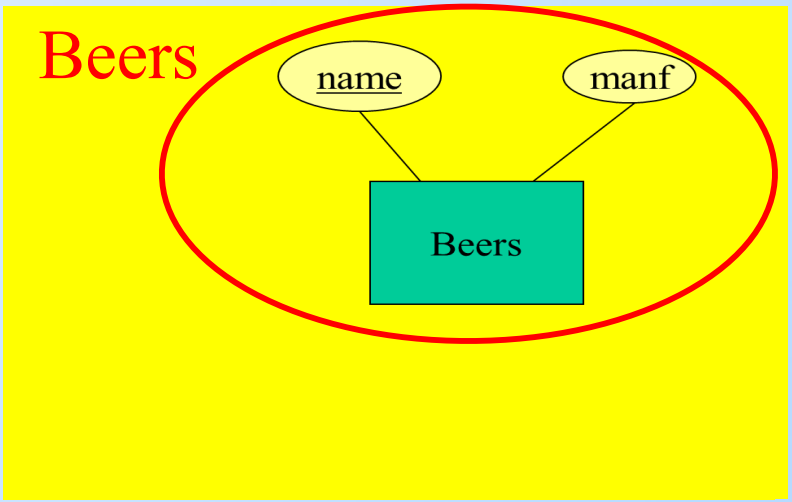
What *to* How



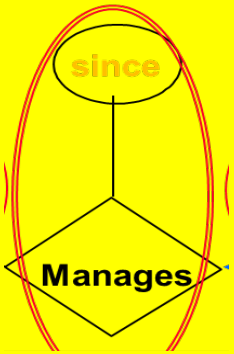

```
TABLE format

TABLE Customer (
                CustomerID          CHAR(20),
                Cname               CHAR(20),
                Address             CHAR(20),
                State               CHAR(20),
                City                CHAR(20),
                PostalCode          CHAR(15),
                Email               CHAR(20),
                UserName            CHAR(20),
                Password            CHAR(20),
                PRIMARY KEY         (CustomerID))

TABLE ProductLine (LineID           CHAR(30),
                   Lname            CHAR(30),
                   PRIMARY KEY(LineID))

TABLE Order (
              OrderID               CHAR (20),
              orderDate             DATE,
                CustomerID                CHAR(20),
              PRIMARY KEY       (OrderID),
              FOREIGN KEY       (CustomerID) REFRENCES Customer)

TABLE Product (
                ProductID               CHAR(20),
                Pname                   CHAR(20),
                Finish                  CHAR(20),
                StandartPrice.          FLOAT,
                Photo                   IMAGE,
                LineID                  CHAR(20),
                PRIMARY KEY             (ProductID),
                FOREIGN KEY (LineID) REFRENCES ProductLine)

TABLE OrderLine (
                  Quantity                  INTEGER,
                  SalePrice.                FLOAT,
                  ProductID                       CHAR(20),
                  OrderID                   CHAR(20),
                  PRIMARY KEY                     (ProductID, OrderID),
                  FOREIGN KEY (ProductID) REFERENCES Product,
                  FOREIGN KEY (ProductLineID) REFRENCES Order)
```

From ER Diagrams to Relations

- Entity set -> Relation.
  - attributes -> attributes.



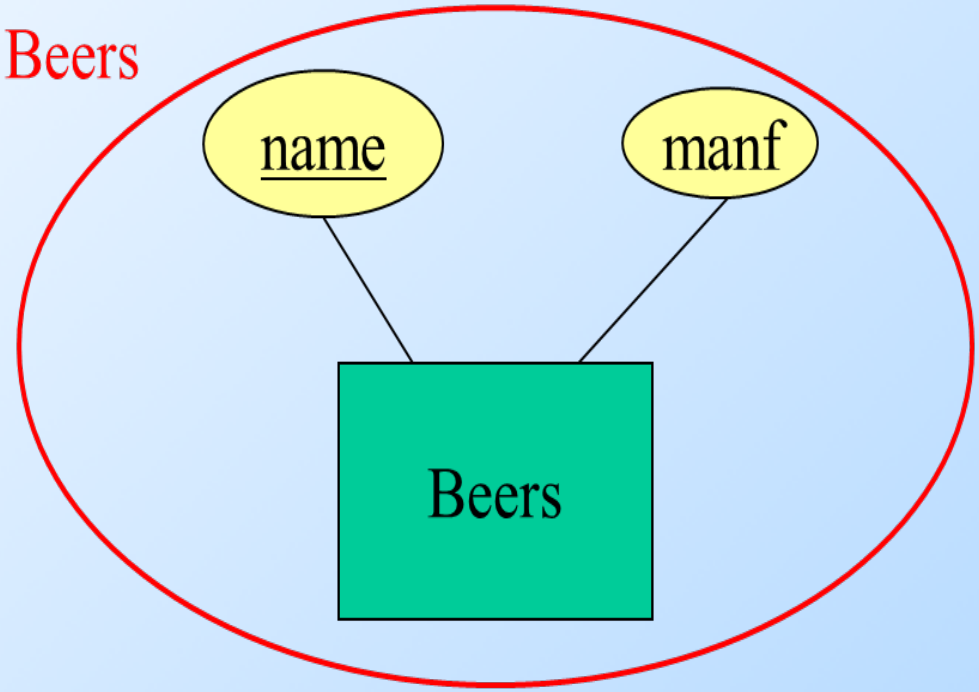
- Relationship set -> Relation whose attributes are:
  - The keys of the connected Entity sets.
  - attributes of the Relationship set itself.



Manages

How many attributes will Manages have? **3!**

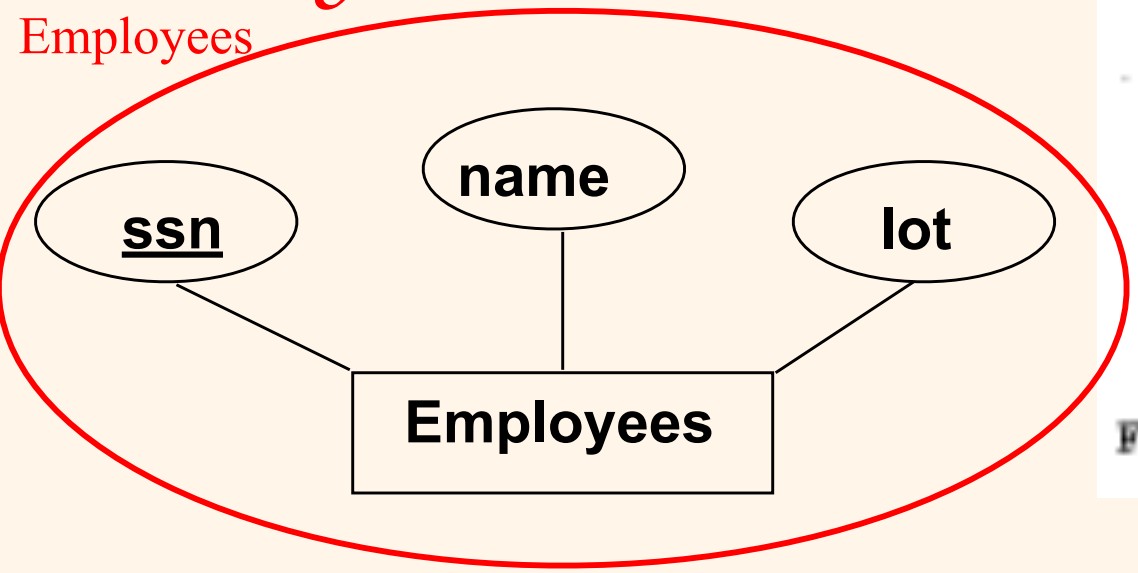
Entity Set -> Relation



Relation: Beers(name, manf)

Logical DB Design: ER to Relational

Entity Sets to Relations:



ssn	name	lot
123-22-3666	Attishoo	48
231-31-5368	Smiley	22
131-24-3650	Smethurst	35

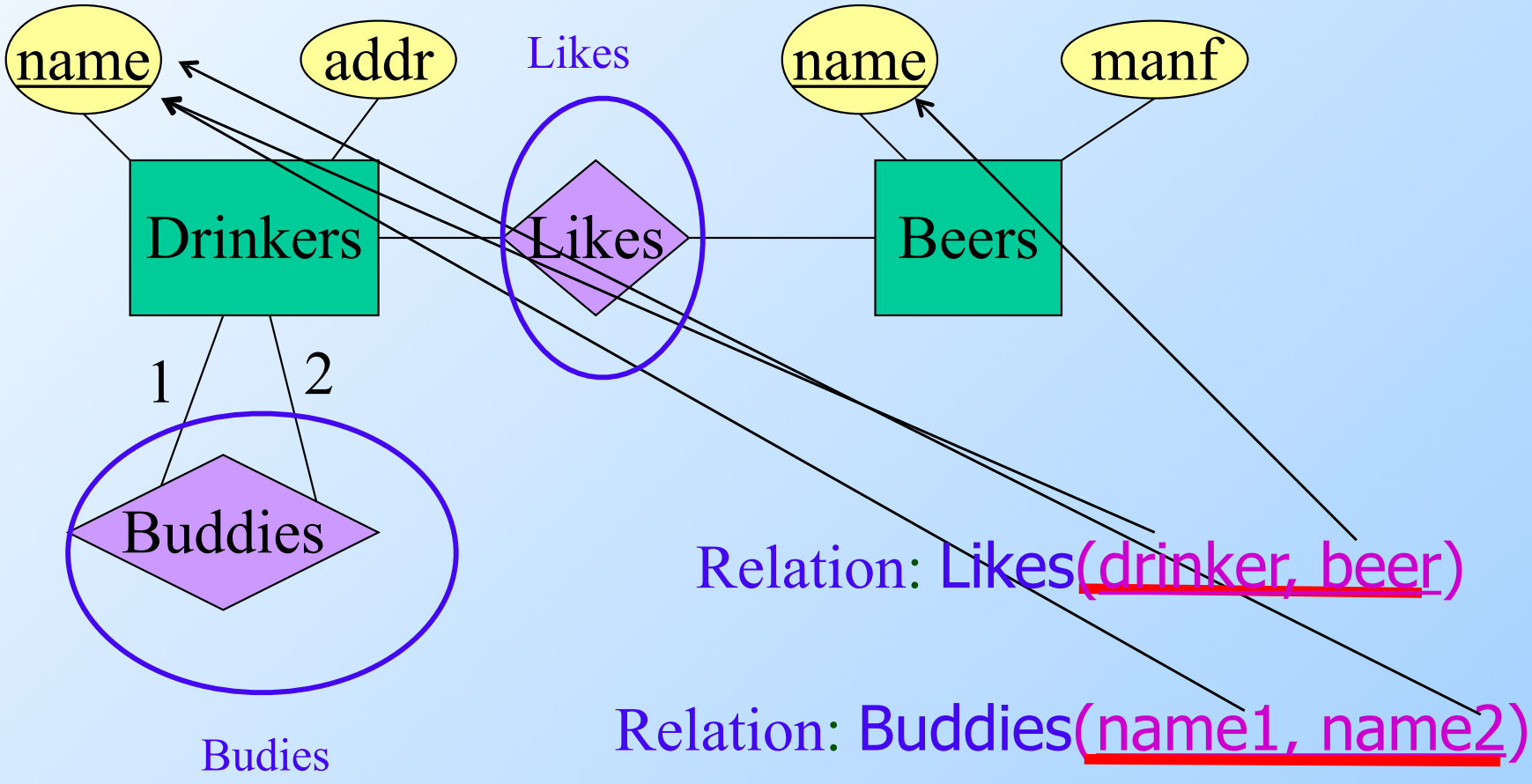
Figure 3.9 An Instance of the Employees Entity Set

Relation: Employees(ssn, name, lot)

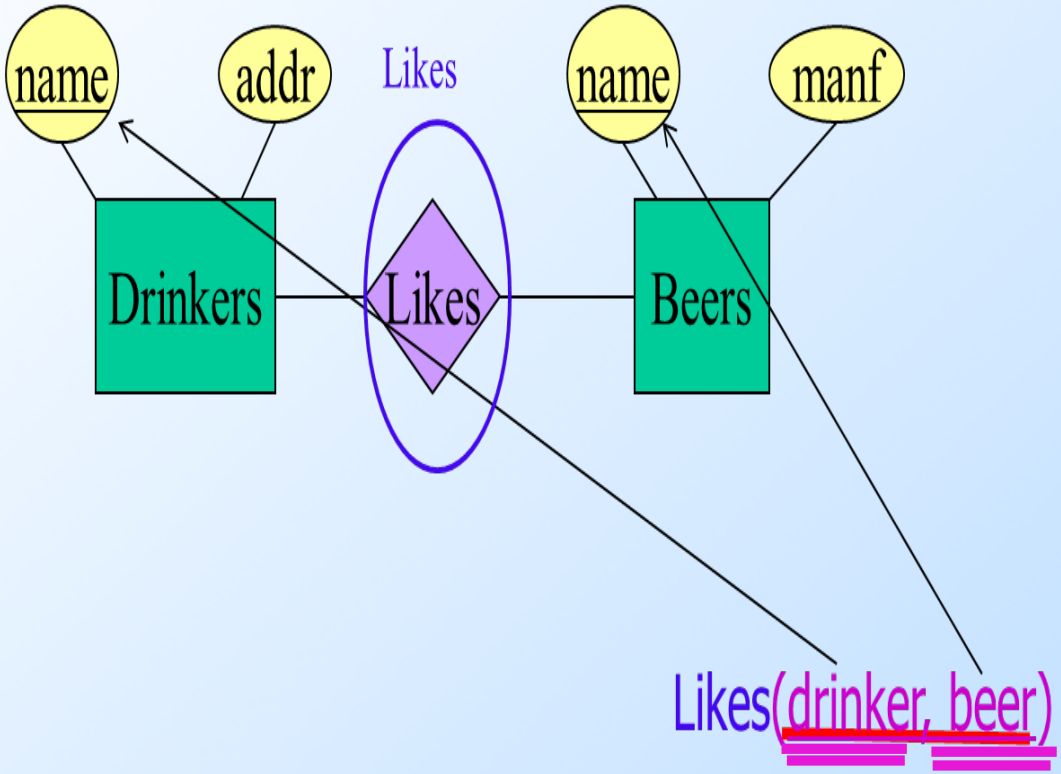
```
TABLE Employees ( ssn      CHAR(11),
                  name     CHAR(30),
                  lot      INTEGER,
                  PRIMARY KEY (ssn) )
```

preferred

Relationship set -> Relation



Relationship set -> Relation

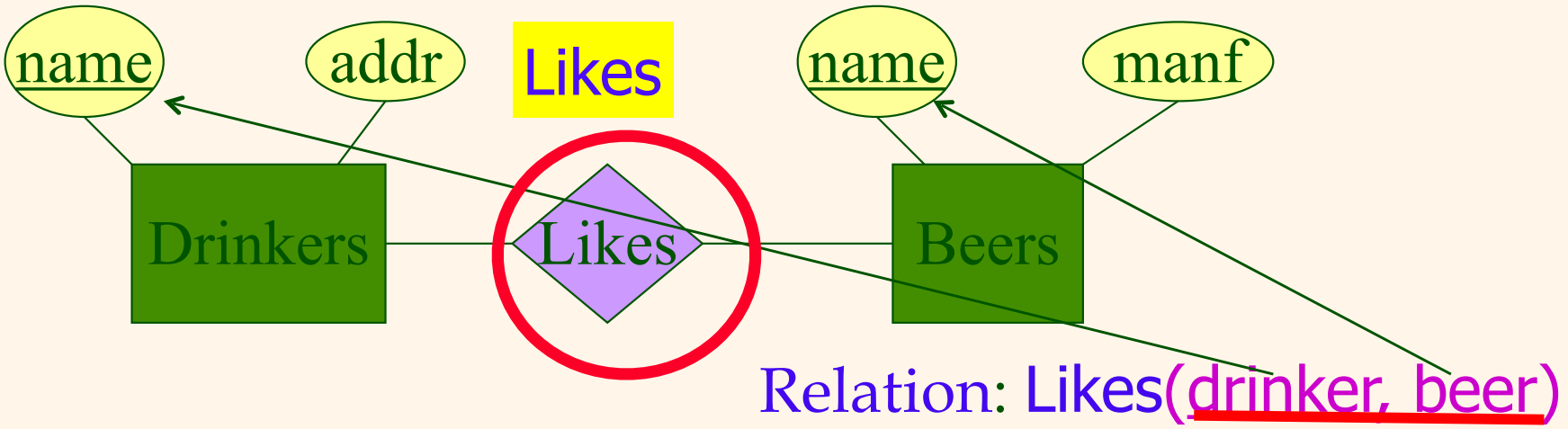


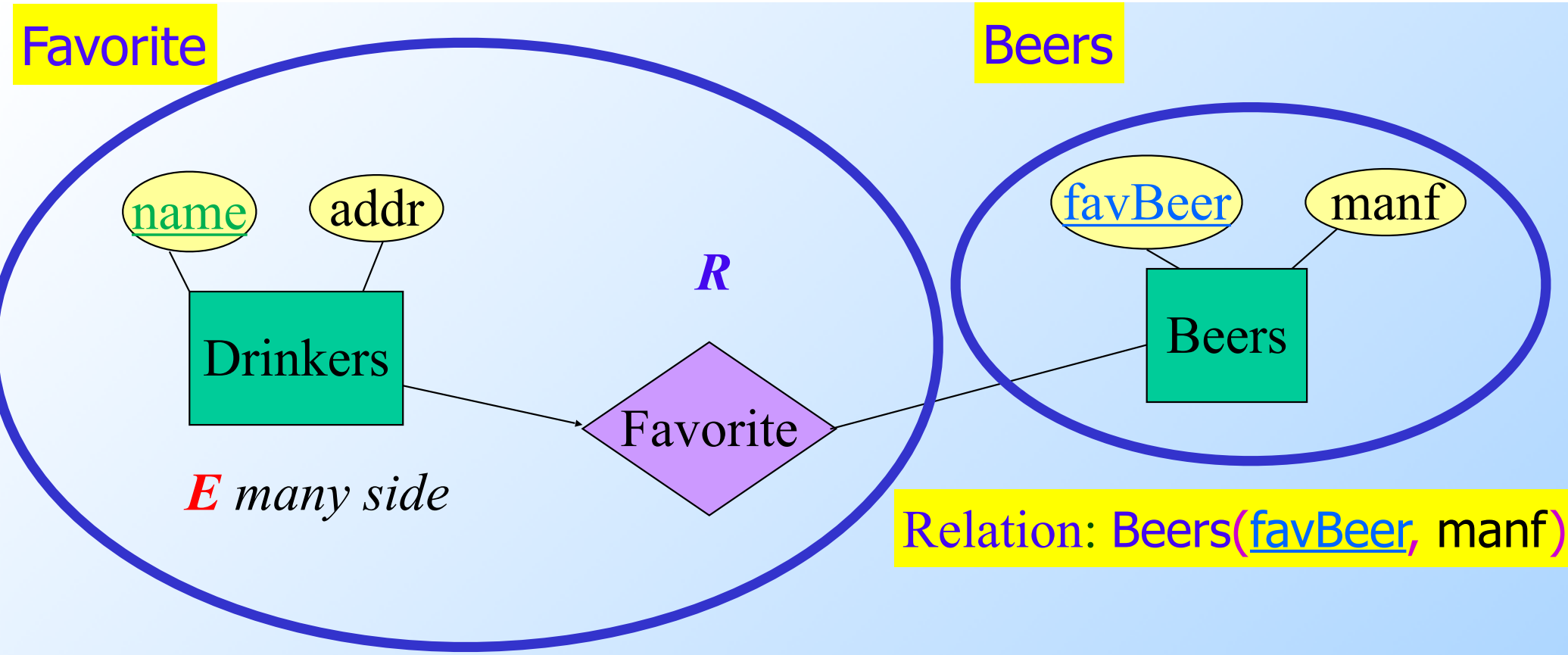
- Relationship set -> Relation whose attributes are:
  - The keys of the connected Entity sets. — PK
  - attributes of the Relationship set itself. — FK

Logical DB Design: ER to Relational

Relationship sets to Relations:

- In translating a 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 *key* for the **Relation**.
  - All descriptive attributes (attributes of the Relationship set!).





Relation: Beers(favBeer, manf)

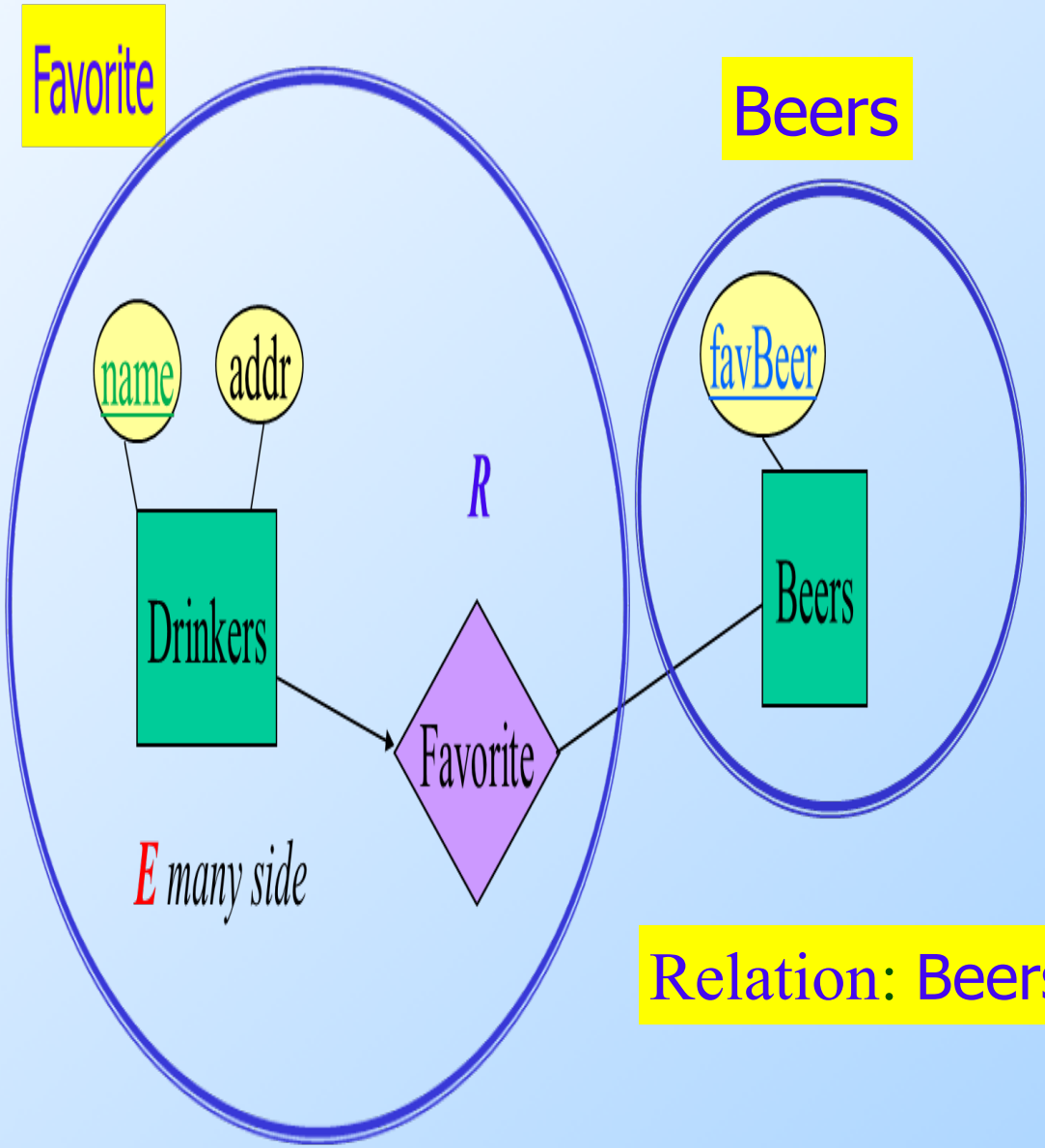
Relation: Favorite(name, addr, favBeer)

2 **Relations**!

Combine into **one Relation**:

1. The **Relation** for an Entity Set **E**
2. The **Relations** for many-one Relationships **R** of which **E** is the "many"

How many attributes will the resulting **Relation** Favorite has?



Relation: Beers(favBeer, manf)

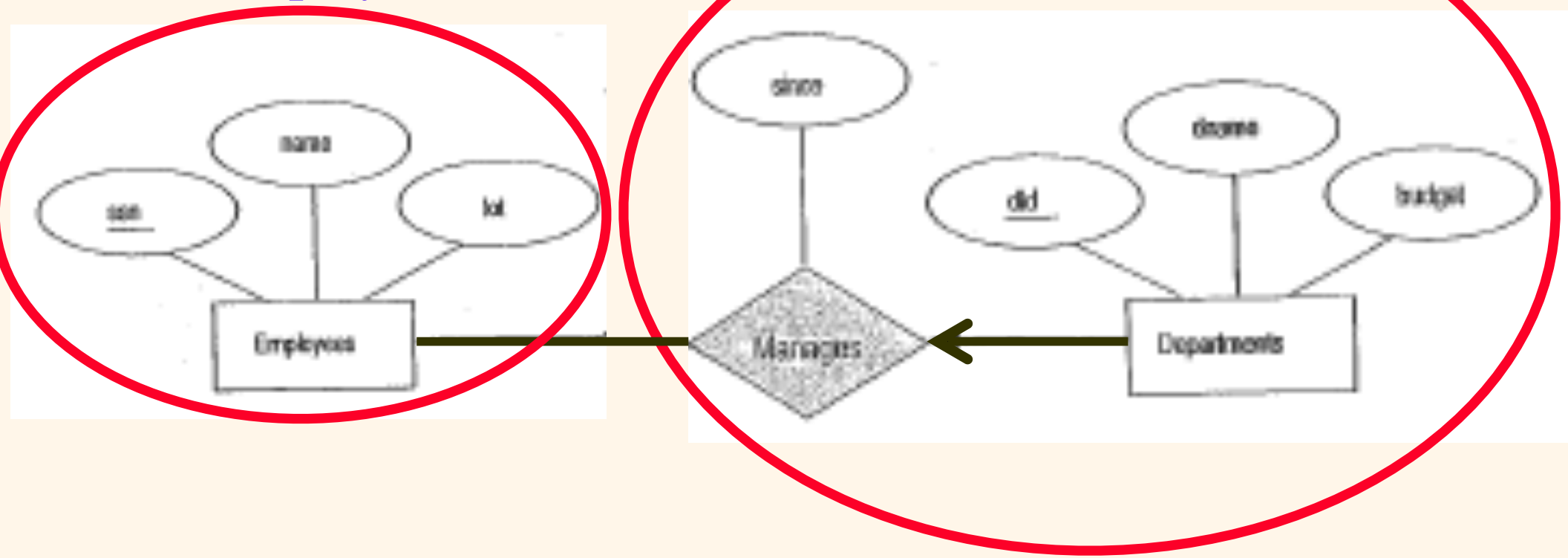
Relation: Favorite(name, addr, favBeer)

Translating ER Diagrams with Key Constraints

- Since *each Department has a unique Manager*, we could instead combine **Manages** and **Departments**.

Dept_Mgr

Employees



How many **Relations** we will have? 2

Translating ER Diagrams with Key Constraints

Dept_Mgr

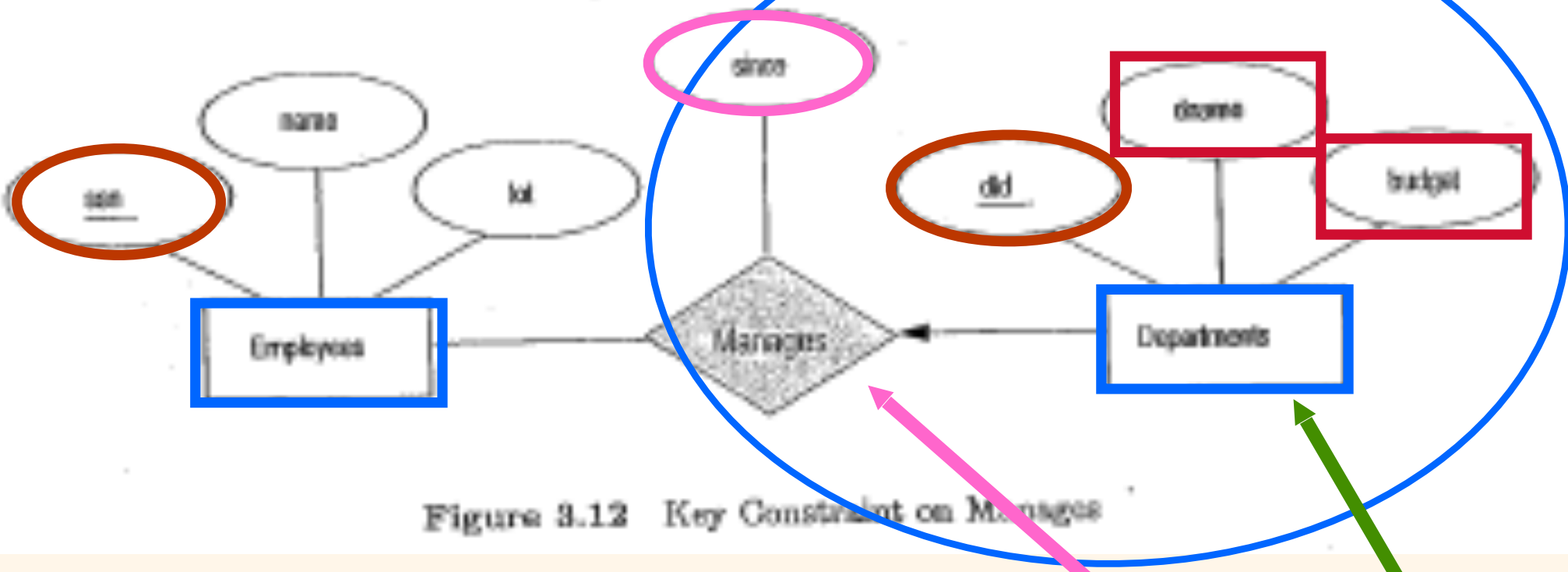
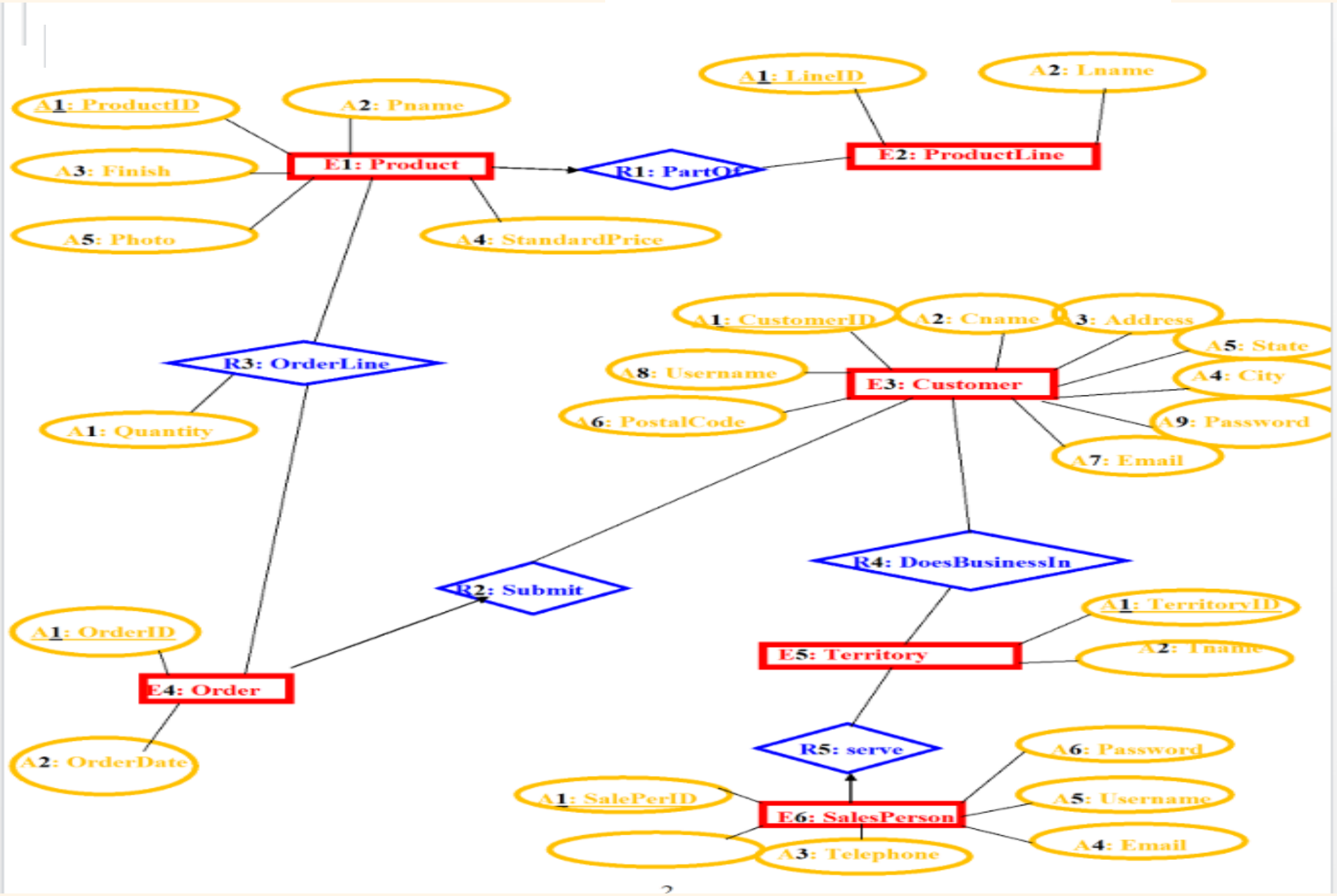


Figure 3.12 Key Constraint on Manages

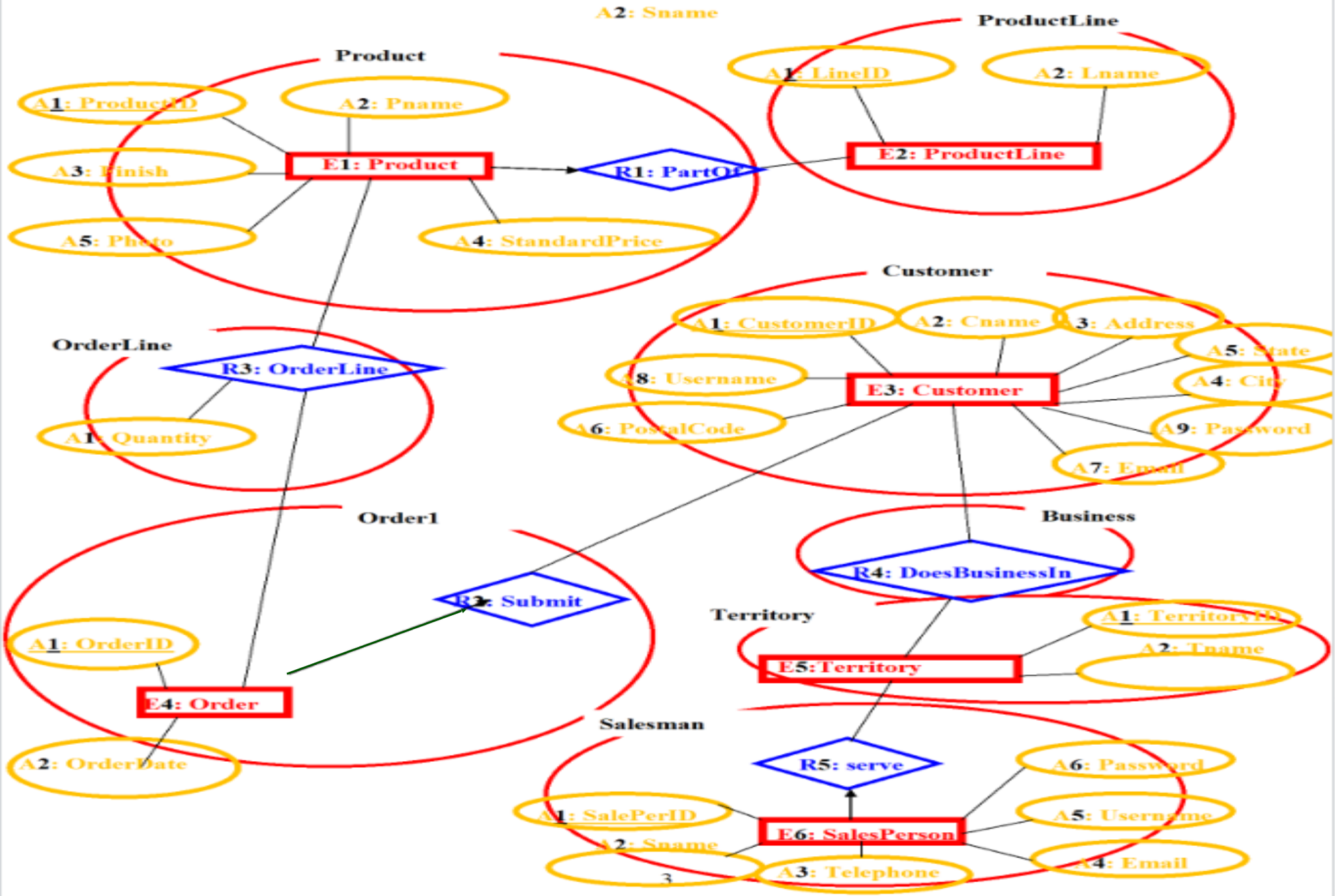
```
TABLE Dept_Mgr ( did        INTEGER,
                 dname      CHAR(20),
                 budget     REAL,
                 ssn        CHAR(11),
                 since      DATE,
                 PRIMARY KEY (    ),
                 FOREIGN KEY (    ) REFERENCES Employees )
```

The idea is to include information about the Relationship set in the **Relation** corresponding to the Entity set with the key, taking advantage of the key constraint!

WHAT - ER Diagram



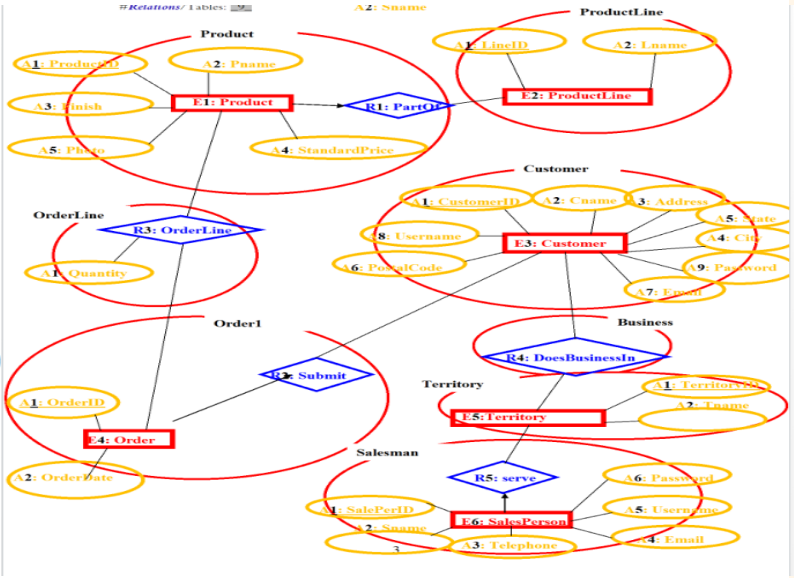
WHAT - ER Diagram to HOW – Relations Step 1



WHAT - ER Diagram to HOW – Relations Step 2

TABLE format

```
TABLE Customer (
                    CustomerID          CHAR(20),
                    Cname               CHAR(20),
                    Address             CHAR(20),
                    State               CHAR(20),
                    City                CHAR(20),
                    PostalCode          CHAR(15),
                    Email               CHAR(20),
                    UserName            CHAR(20),
                    Password            CHAR(20),
                    PRIMARY KEY         (CustomerID))

TABLE ProductLine (LineID               CHAR(30),
                    Lname               CHAR(30),
                    PRIMARY KEY(LineID))

TABLE Order (
                OrderID                 CHAR (20),
                orderDate               DATE,
                  CustomerID                CHAR(20),
                 PRIMARY KEY    (OrderID),
                 FOREIGN KEY    (CustomerID) REFRENCES Customer)

TABLE Product (
                    ProductID           CHAR(20),
                    Pname               CHAR(20),
                    Finish              CHAR(20),
                    StandartPrice.      FLOAT,
                    Photo               IMAGE,
                    LineID              CHAR(20),
                    PRIMARY KEY         (ProductID),
                    FOREIGN KEY (LineID) REFRENCES ProductLine)

TABLE OrderLine (
                    Quantity            INTEGER,
                    SalePrice.          FLOAT,
                    ProductID               CHAR(20),
                    OrderID             CHAR(20),
                    PRIMARY KEY             (ProductID, OrderID),
                    FOREIGN KEY (ProductID) REFERENCES Product,
                    FOREIGN KEY (ProductLineID) REFRENCES Order)
```



```
TABLE Territory (
                TerritoryID          CHAR(30),
                Tname.        CHAR(30),
                PRIMARY KEY (TerritoryID))

TABLE Business (
                CustomerID        CHAR(30),
                TerritoryID       CHAR(30),
                PRIMARY KEY (CustomerID, TerritoryID),
                FOREIGN KEY (CustomerID) REFERENCES Customer,
                FOREIGN KEY (TerritoryID) REFRENCES Territory)

TABLE SalesMan (
                SalePerID             CHAR(30),
                Sname                 CHAR(30),
                Telephone             CHAR(15),
                  Email                       CHAR(30),
                Username              CHAR(30),
                Password               CHAR(30),
                TerritoryID           CHAR(30),
                PRIMARY KEY           (SalePerID),
                FOREIGN KEY (TerritoryID) REFERENCES Territory)

TABLE PriceUpdates (
                 Uname                CHAR(30),
                  changeDate          DATE,
                  OldPrice            FLOAT,
                  NewPrice            FLOAT,
                  PRIMARY KEY         (Uname, OldPrice))
```

TA, Alvaro Urtaza (A – L).

TA, Jordan, Yu (M – Z).

Please compare CANVAS vs. TEAMS Attendance.
Print screens of students in CANVAS but not in the TEAMS meeting.
(1.29.2024 Attendance X missing LastName.docx)

ERD - WHAT to Relational - HOW

5. SET 1 - 4:ERD to RELATIONAL

0% 0%

ENTITY SETS - WHAT to Relational - HOW

5.1 Implementing entities

0% 0%

RELATIONSHIP SETS - WHAT to Relational - HOW

5.2 Implementing relationships

0% 0%

ATTRIBUTES - WHAT to Relational **ATTRIBUTES** - HOW

5.3 Implementing attributes

0% 0%

Selecting primary keys

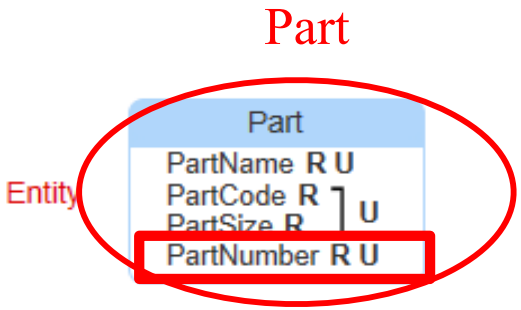
In the first step of logical design, entities become tables and attributes become columns. As tables and columns are specified, primary keys are selected. Primary keys must be unique and required (not NULL). Primary keys should also be:

- *Stable*. Primary key values should not change. When a primary key value changes, statements that specify the old value must also change. Furthermore, the new primary key value must cascade to matching foreign keys.
- *Simple*. Primary key values should be easy to type and store. Small values are easy to specify in an SQL WHERE clause and speed up query processing. Ex: A 2-byte integer is easier to type and faster to process than a 15-byte character string.
- *Meaningless*. Primary keys should not contain descriptive information. Descriptive information occasionally changes, so primary keys containing descriptive information are unstable.

Stable, simple, and meaningless primary keys are desirable but not necessary. Occasionally, these guidelines may be violated.

In table diagrams, a bullet (●) indicates a primary key column.

PARTICIPATION ACTIVITY 5.1.1: Selecting primary keys.



5.1 Implementing entities

ENTITY SETS

Selecting primary keys

In the first step of the logical design phase, each entity becomes a table and each attribute becomes a column. Table and columns are reviewed and revised in subsequent logical design steps.

As tables and columns are specified, primary keys are selected. Primary keys must be unique and not NULL. A non-NULL column is equivalent to a required attribute, so primary keys correspond to unique and required attributes. In addition, primary keys should be:

- **Stable**. Primary key values should not change. Unstable primary keys cause database management problems. When a primary key value changes, statements that specify the old value must also change, and the new primary key value must cascade to foreign keys.
- **Simple**. Primary key values should be easy to type and store. Small values are easy to specify in an SQL WHERE clause and speed up query processing. Ex: A 2-byte integer is easier to type and faster to process than a 15-byte character string.
- **Meaningless**. Primary keys should not contain descriptive information. Descriptive information occasionally changes, so primary keys containing descriptive information are unstable.

On table diagrams, solid bullets denote attributes that become primary keys.

PARTICIPATION ACTIVITY | 5.1.2: Selecting primary keys.

1) A _____ attribute becomes a column that is never NULL.

required

Check | Show answer

required

NULL stands for unknown or inapplicable. Columns with NULL values correspond to optional attributes. Columns that are never NULL correspond to required attributes.

?????

Entity Set on the One Side – WHAT to Relational Model – HOW

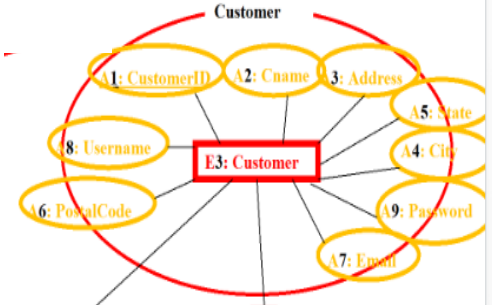


TABLE format

```
TABLE Customer (
        CustomerID      CHAR(20),   A1
        Cname           CHAR(20),   A2
        Address         CHAR(20),   A3
        State           CHAR(20),   A4
        City            CHAR(20),   A5
        PostalCode      CHAR(15),   A6
        Email           CHAR(20),   A7
        UserName        CHAR(20),   A8
        Password        CHAR(20),   A9
        PRIMARY KEY     (CustomerID))
```

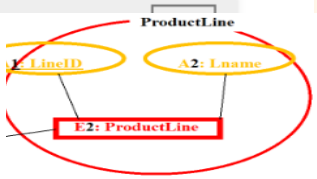
Implementing strong entities

ENTITY SETS – on the 1 side

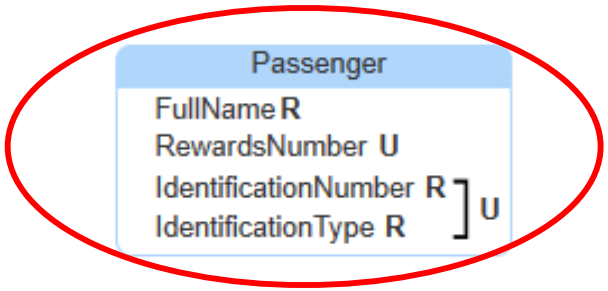
A strong entity becomes a ***strong table***. The primary key must be unique and required, and should be stable, simple, and meaningless.

Simple primary keys are best for strong tables. If no simple primary key is available, a composite primary key may be selected. Alternatively, the database designer may create an artificial primary key. An ***artificial key*** is a simple primary key created by the database designer. Usually artificial keys are integers, generated automatically by the database as new rows are inserted to the table. Artificial keys are stable, simple, and meaningless.

PARTICIPATION ACTIVITY | 5.1.3: Implementing strong entities.



Passenger



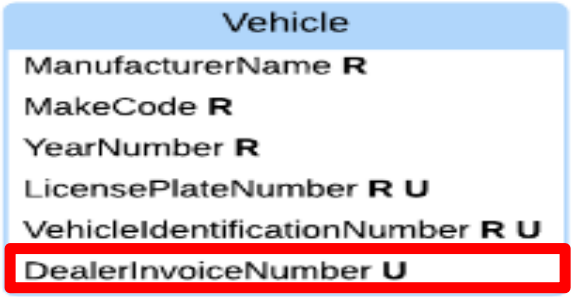
Implementing strong entities

A strong entity becomes a ***strong table***. The primary key must be unique and required, and should be stable, simple, and meaningless.

Simple primary keys are best for strong tables. If no simple primary key is available, a composite primary key may be selected. Alternatively, the database designer may create an artificial primary key. An ***artificial key*** is a simple primary key created by the database designer. Usually artificial keys are integers, generated automatically by the database as new rows are inserted to the table. Artificial keys are stable, simple, and meaningless.

PARTICIPATION ACTIVITY | 5.1.4: Primary key of independent entity.

Vehicle is a strong entity:



1) DealerInvoiceNumber

- ○ True
- ◉ False

Correct

DealerInvoiceNumber is not followed by an R and therefore is optional. Optional attributes become columns with NULL values allowed, which cannot be primary keys.

?????

Implementing weak entities

A weak entity becomes a ***weak table***. A weak table has a foreign key that references the identifying table and implements the identifying relationship.

The primary key depends on the cardinality of the identifying relationship:

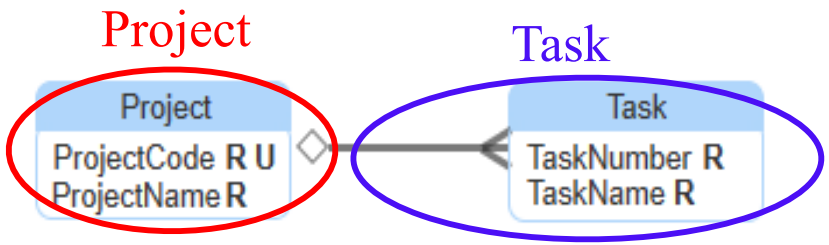
- Usually, the weak entity is plural. The primary key is the composite of the foreign key and another column.
- Occasionally, the weak entity is singular. The primary key is the foreign key only.

The foreign key usually has the following referential integrity actions:

- Cascade on primary key update and delete
- Restrict on foreign key insert and update

In table diagrams, an arrow indicates a foreign key. The arrow starts at the foreign key and points to the table containing the referenced primary key.

PARTICIPATION ACTIVITY 5.1.5: Implementing weak entities.



Implementing weak entities

relationship.

The primary key depends on the cardinality of the identifying relationship:

- Usually, the weak entity is plural. The primary key is the composite of the foreign key and another column.
- Occasionally, the weak entity is singular. The primary key is the foreign key only.

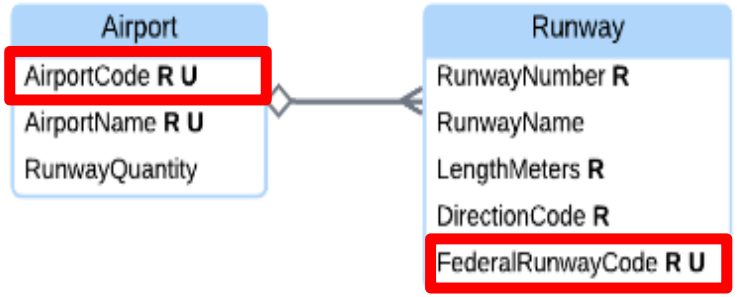
The foreign key usually has the following referential integrity actions:

- Cascade on primary key update and delete
- Restrict on foreign key insert and update

In table diagrams, an arrow indicates a foreign key. The arrow starts at the foreign key and points to the table containing the referenced primary key.

PARTICIPATION ACTIVITY 5.1.7: Implementing weak entities.

Runway is a weak entity, identified by the strong entity Airport:



AirportCode is the primary key of the Airport table. At each airport, runway number is unique.

Which of the following is a good primary key for the Runway table?

4) FederalRunwayCode

- (●) True
- ( ) False

Correct

Usually, a weak table primary key is the composite of the identifying table primary key and another column. However, the database designer may prefer a simple primary key. If so, FederalRunwayCode is unique, required, simple, and stable, and hence a good alternative.

?????

Implementing supertype and subtype entities

A supertype entity becomes a ***supertype table***. A supertype entity that has an identifying attribute is implemented like a strong entity. A supertype entity that has an identifying relationship is implemented like a weak entity.

A subtype entity becomes a ***subtype table***.

- The primary key is identical to the supertype primary key.
- The primary key is also a foreign key that references the supertype primary key.

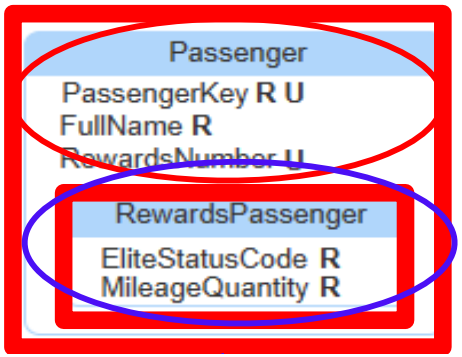
The foreign key usually has the following referential integrity actions:

- Cascade on primary key update and delete
- Restrict on foreign key insert and update

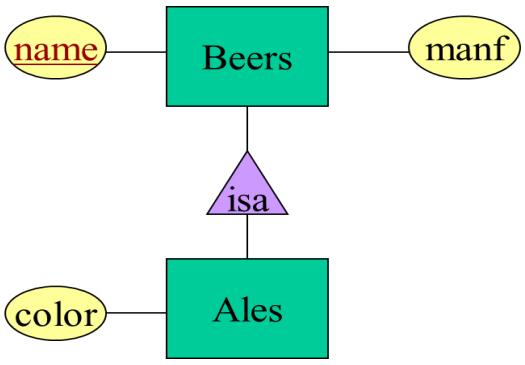
The foreign key implements the IsA relationship between subtype and supertype entities.

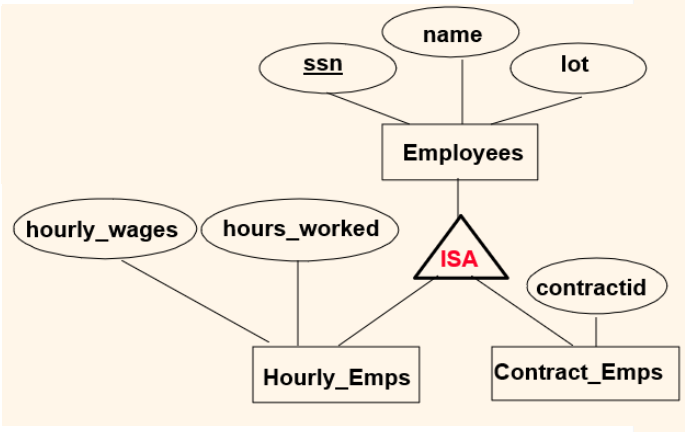
PARTICIPATION ACTIVITY 5.1.8: Implementing subtype entities.

Passenger



IsA





Implementing subtype entities

A subtype entity becomes a ***subtype table*** and is implemented as follows:

- The primary key is identical to the supertype primary key.
- The primary key is also a foreign key that references the supertype primary key.

The foreign key implements the 'IsA' dependency relationship. Foreign keys that implement dependency relationships usually have the following referential integrity actions:

- Cascade on primary key update and delete
- Restrict on foreign key insert and update

Referential integrity actions are described elsewhere in this material.

On table diagrams, open bullets denote attributes that become foreign keys.

PARTICIPATION ACTIVITY 5.1.6: Implementing subtype entities.

1) The subtype table primary key is identical to the _____ table primary key.

supertype

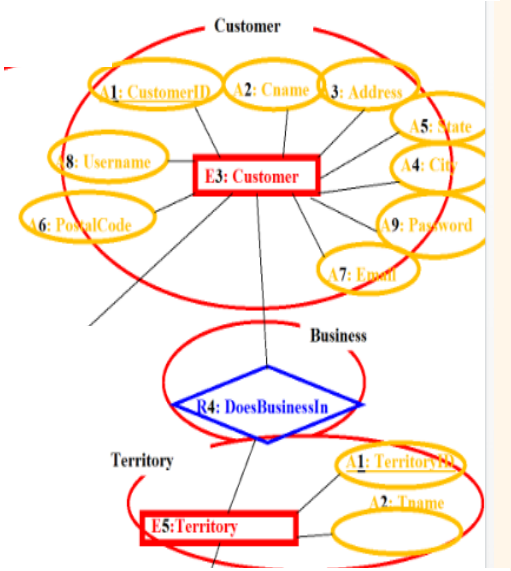
Check Show answer

supertype

Since subtype entity instances are always supertype entity instances, subtype tables have the same primary key as the supertype table.

?????

Relationship Set (Many to Many) WHAT to Relational – HOW




```
TABLE Business (
        CustomerID      CHAR(30),
        TerritoryID     CHAR(30),
        PRIMARY KEY (CustomerID, TerritoryID),
        FOREIGN KEY (CustomerID) REFERENCES Customer,
        FOREIGN KEY (TerritoryID) REFRENCES Territory)
```

5.2 Implementing relationships

RELATIONSHIP SETS Many to Many

Implementing relationships

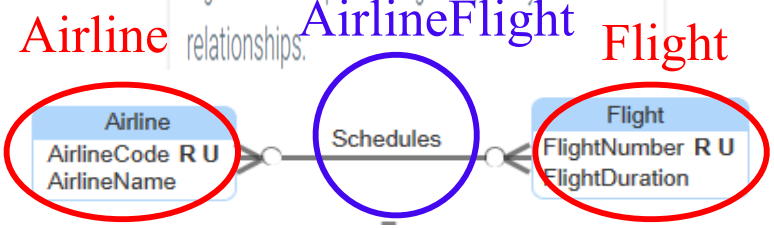
Implementing many-many relationships

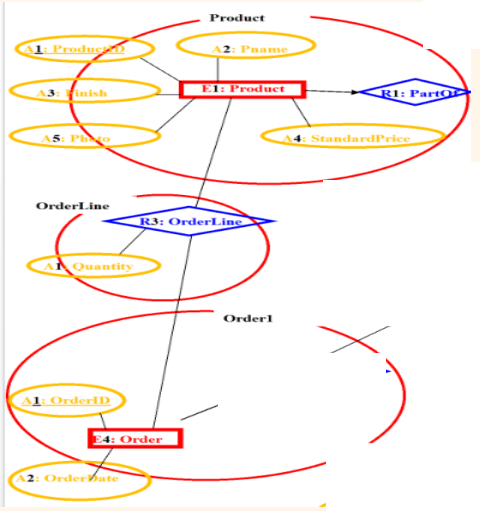
A many-many relationship becomes a new table:

- The new table contains two foreign keys, referring to the primary keys of the related tables.
- The primary key of the new table is the composite of the two foreign keys.
- The new table is dependent on the related tables, so primary key cascade and foreign key restrict rules are specified.
- The new table name consists of the related table names with an optional qualifier in between. The qualifier is derived from the relationship name and clarifies the meaning of the table.

In the figure below, the many-many relationship Airline-Schedules-Flight becomes the new table AirlineFlight. The primary keys from Airline and Flight become foreign keys in AirlineFlight, and the composite key (AirlineCode, FlightNumber) becomes AirlineFlight's primary key.

Figure 5.2.1: Implementing many-many relationships.





5.2 Implementing relationships

RELATIONSHIP SETS One to One

Implementing many-many relationships

A many-many relationship becomes a new table:

- The new table contains two foreign keys, referring to the primary keys of the related tables.
- The primary key of the new table is the composite of the two foreign keys.
- The new table is dependent on the related tables, so primary key cascade and foreign key restrict rules are specified.
- The new table name consists of the related table names with an optional qualifier in between. The qualifier is derived from the relationship name and clarifies the meaning of the table.

PARTICIPATION ACTIVITY 5.2.5: Implementing many-many relationships.

A passenger can have many credit cards on file, and a single credit card can be issued in the name of many passengers:

Refer to the new table implementing the many-many relationship Owns.

1) What is the primary key of the new table?

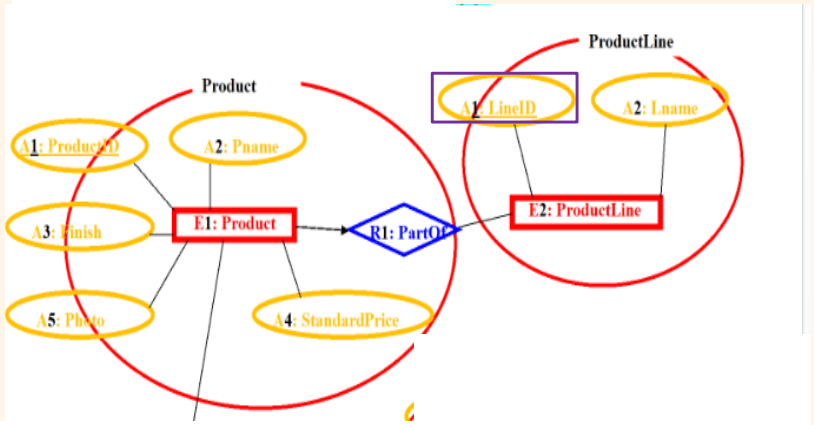
- ○ CreditCardNumber
- ◉ (PassengerNumber, CreditCardNumber)
- ○ The composite of PassengerNumber, CreditCardNumber, and a third column of the new table.

Correct

The primary key of a table implementing a many-many relationship is the composite of primary keys of the related tables.

?????

Relationship Set (Many to One) WHAT to Relational – HOW




```
TABLE Product (
        ProductID          CHAR(20),      A1
        Pname              CHAR(20),      A2
        Finish             CHAR(20),      A3
        StandartPrice.     FLOAT,         A4
        Photo              IMAGE,         A5
        LineID             CHAR(20),
        PRIMARY KEY        (ProductID),
        FOREIGN KEY (LineID) REFRENCES ProductLine)
```

Implementing relationships

Implementing many-one relationships

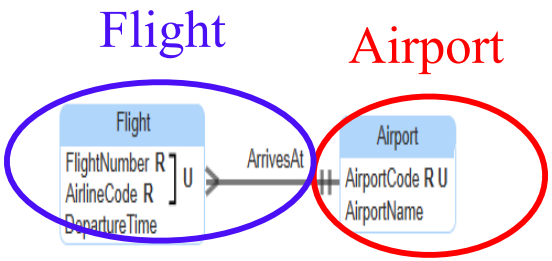
The 'implement entities' step converts identifying relationships into foreign keys. The 'implement relationships' step converts all other relationships into foreign keys or tables.

A many-one or one-many relationship becomes a foreign key:

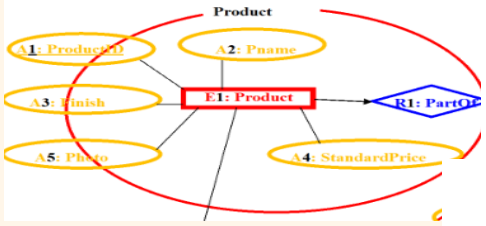
- The foreign key goes in the table on the 'many' side and refers to the table on the 'one' side.
- If the entity on the 'one' side is required, the foreign key column is also required.

The foreign key name is the name of the referenced primary key, with an optional prefix. The prefix is usually derived from the relationship name and clarifies the meaning of the foreign key.

PARTICIPATION ACTIVITY 5.2.1: Implementing many-one relationships.



RELATIONSHIP SETS Many to One




```
TABLE Product (
        ProductID          CHAR(20),     A1
        Pname              CHAR(20),     A2
        Finish             CHAR(20),     A3
        StandartPrice.     FLOAT,        A4
        Photo              IMAGE,        A5
        LineID             CHAR(20),
        PRIMARY KEY        (ProductID),
        FOREIGN KEY (LineID) REFRENCES ProductLine)
```

5.2 Implementing relationships

RELATIONSHIP SETS Many to One

Implementing many-one relationships

The 'implement relationships' step converts relationships into keys or tables, depending on relationship cardinality.

A many-one or one-many relationship becomes a foreign key:

- The foreign key goes in the table on the 'many' side of the relationship.
- The foreign key refers to the primary key on the 'one' side.
- The foreign key name is the primary key name with an optional prefix. The prefix is derived from the relationship name and clarifies the meaning of the foreign key.

PARTICIPATION ACTIVITY 5.2.2: Implementing many-one relationships.

Refer to the following relationship:

The primary key of the Aircraft table is AircraftCode. The primary key of the Flight table is (FlightNumber, AirlineCode).

1) In which table is the foreign key placed?

- ○ Aircraft
- ◉ Flight
- ○ The table with fewer rows

Correct

The foreign key goes in the table derived from the entity on the 'many' side of Aircraft-IsAssignedTo-Flight, which is Flight.

ABLE Product (

ProductID	CHAR(20),
Pname	CHAR(20),
Finish	CHAR(20),
StandartPrice.	FLOAT,
Photo	IMAGE,
LineID	CHAR(20),
PRIMARY KEY	(ProductID),
FOREIGN KEY (LineID) REFRENCES ProductLine)	

?????

5.2 Implementing relationships

RELATIONSHIP SETS One to One

Present Note

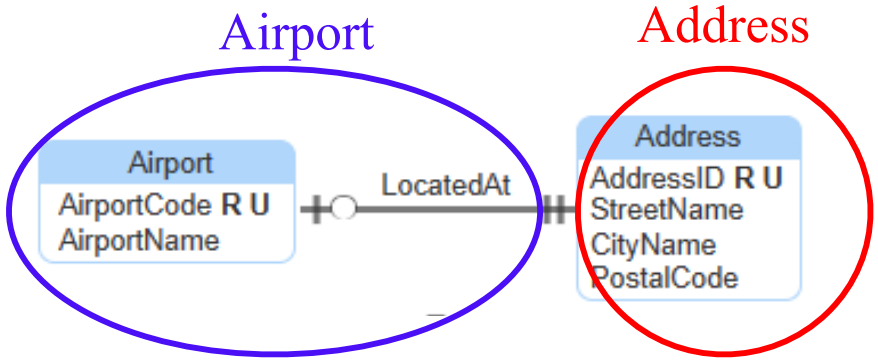
Implementing one-one relationships

A one-one relationship becomes a foreign key. The foreign key can go in the table on either side of the relationship. Usually, the foreign key is placed in the table with fewer rows, to minimize the number of NULL values.

- The foreign key refers to the table on the opposite side of the relationship.
- The foreign key column is unique.
- If the entity on the opposite side of the relationship is required, the foreign key column is also required.

The foreign key name is the name of the referenced primary key, with an optional prefix. The prefix is usually derived from the relationship name and clarifies the meaning of the foreign key.

PARTICIPATION ACTIVITY 5.2.3: Implementing one-one relationships.



5.2 Implementing relationships

RELATIONSHIP SETS One to One

Implementing one-one relationships

A one-one relationship becomes a foreign key:

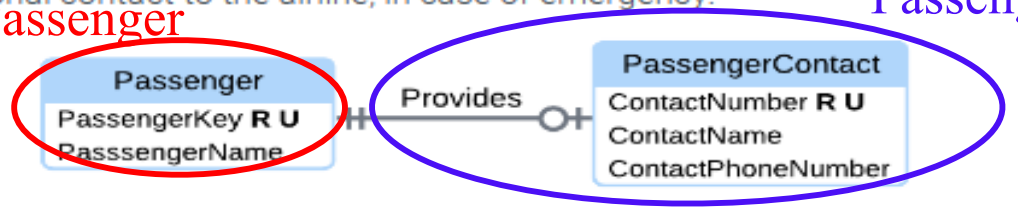
- The foreign key can go in the table on either side of the relationship. Usually, the foreign key is placed in the table with fewer rows to minimize the number of NULL values.
- The foreign key refers to the primary key on the opposite side of the relationship.
- The foreign key name is the primary key name with an optional prefix. The prefix is derived from the relationship name and clarifies the meaning of the foreign key.

PARTICIPATION ACTIVITY 5.2.4: Implementing one-one relationships.

Some passengers provide a personal contact to the airline, in case of emergency:

Passenger

PassengerContact



The primary key of the Passenger table is PassengerKey. The primary key of the PassengerContact table is ContactNumber.

1) In which table is the foreign key placed?

- ○ Passenger
- ◉ PassengerContact
- ○ Cannot be determined from the diagram.

Correct

All contacts are provided by a passenger, but not all passengers provide a contact, so Contact has fewer rows. For one-one relationships, the foreign key is usually placed in the table with fewer rows.

?????

5.3 Implementing attributes

Implementing plural attributes

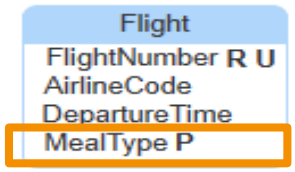
In the 'implement entities' step, entities become tables and attributes become columns. Singular attributes remain in the initial table, but plural attributes move to a new weak table:

- The new table contains the plural attribute and a foreign key referencing the initial table.
- The primary key of the new table is the composite of the plural attribute and the foreign key.
- The new table is identified by the initial table, so primary key cascade and foreign key restrict rules are specified.
- The new table name consists of the initial table name followed by the attribute name.

If a plural attribute has a small, fixed maximum, the plural attribute can be implemented as multiple columns in the initial table. However, implementing plural attributes in a new table simplifies queries and is usually a better solution.

PARTICIPATION ACTIVITY 5.3.1: Implementing plural attributes.

Standard Design



5.3 Implementing attributes

Implementing plural attributes

Attributes can be singular or plural:

- Each entity instance has at most one singular attribute instance.
- Each entity instance can have many plural attribute instances.

In the 'implement entities' step, entities become tables and attributes become columns. Singular attributes remain in the initial table, but plural attributes move to a new table:

- The new table contains the plural attribute and a foreign key referencing the initial table.
- The primary key of the new table is the composite of the plural attribute and the foreign key.
- The new table is dependent on the initial table, so primary key cascade and foreign key restrict rules are specified.
- The new table name consists of the initial table name followed by the attribute name.

PARTICIPATION ACTIVITY 5.3.2: Implementing plural attributes.

The plural attribute UpdateTime records dates and times when passengers change their seat number:

Booking
FlightNumber **R U**
PassengerKey **R U**
SeatNumber
UpdateTime **P**

Singular attributes are implemented in the Booking table, which has primary key (FlightNumber, PassengerKey). The plural attribute is implemented in a new weak table.

1) What is the name of the new table?

- ○ Booking
- ○ ChangeTimestamp
- ◉ BookingChangeTimestamp

Correct

The name of the new table containing a plural attribute combines the initial table name and the attribute name.

?????

Implementing attribute types

During analysis, a list of standard attribute types is established. During logical design, an SQL data type is defined for each attribute type. Attribute types and the corresponding data types are documented in the glossary.

Each attribute name includes a standard attribute type as a suffix. The attribute type determines the data type of the corresponding column.

PARTICIPATION ACTIVITY 5.3.3: Implementing attribute types.

Start 2x speed

Glossary

```
Attribute Type: Code
Data Type: CHAR(3)
Description: 'Code' is used for
three-character attributes which
identify objects. Characters in
codes are alphabetic or numeric.
Punctuation is not allowed. Examples
are airport codes, such as SFO,
and time zones, such as PDT.

Attribute Type: Name
Data Type: VARCHAR(30)
Description: 'Name' describes
attributes which label information
with free-form names up to 30
characters. Examples include people,
product, and company names.
```

Image did not load.

```
CREATE TABLE Airport (
  AirportCode CHAR(3),
  AirportName VARCHAR(30),
  CountryCode CHAR(3),
  CityName VARCHAR(30)
);
```

Implementing attribute types

During analysis, a list of standard attribute types is established. During logical design, an SQL data type is defined for each attribute type. Attribute types and the corresponding data types are documented in the glossary.

Each attribute name includes a standard attribute type as a suffix. The attribute type determines the data type of the corresponding column.

PARTICIPATION ACTIVITY | 5.3.4: Implementing attribute types.

1) Attribute names always include an attribute type.

- (●) True
- ( ) False

Correct

Every attribute name includes the associated attribute type, preceded by an optional entity name and qualifier.

?????

Implementing attribute cardinality

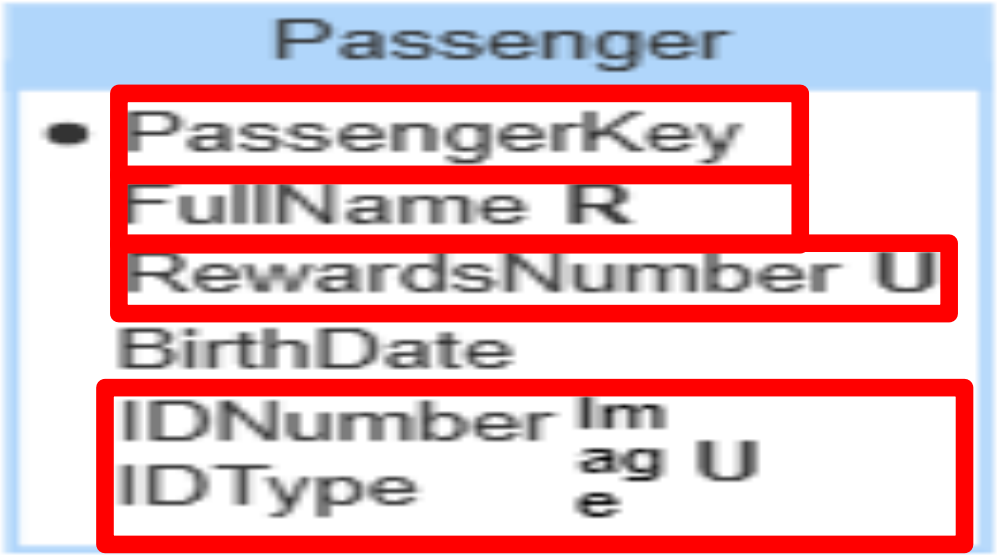


Image did not load.

```
CREATE TABLE Passenger(
    PassengerKey INT PRIMARY KEY,
    FullName VARCHAR(30) NOT NULL,
    RewardsNumber INT UNIQUE
    BirthDate DATE,
    IDNumber INT,
    IDType CHAR(1),
    UNIQUE (IDNumber, IDType)
);
```

Implementing attribute cardinality

Since plural attributes are implemented as singular columns, as described above, all columns are singular. Required or unique attribute become required or unique columns. Like attributes, columns are presumed optional and not unique unless followed by R or U in the ta diagram.

Relationship cardinality determines constraints on foreign key columns:

- If the table *referenced by* the foreign key implements a *required* entity, the column is required.
- If the table *containing* the foreign key implements a *singular* entity, the column is unique.

Table diagrams are implemented as CREATE TABLE statements:

- NOT NULL is specified for required columns.
- UNIQUE is specified for unique columns.
- PRIMARY KEY is specified for primary key columns.

Composite unique columns and composite primary keys cannot be specified in a column definition clause. Composite constraints require an additional clause in the CREATE TABLE statement.

PARTICIPATION ACTIVITY | 5.3.5: Implementing attribute cardinality.

Match the SQL keyword with the cardinality rule.

NOT NULL | no keyword | PRIMARY KEY | UNIQUE

UNIQUE	Maximum cardinality of one. UNIQUE keyword enforces unique attributes and relationships. Unique attributes and relationships have maximum cardinality of one.

?????

Implementing attribute cardinality

Attributes can be unique, required, or optional:

- Each unique attribute instance describes at most one entity instance.
- Each entity instance has at least one required attribute instance.
- Each entity instance can have zero optional attribute instances.

Unique and required attributes are implemented with keywords following the column name in the CREATE TABLE statement:

- UNIQUE is specified on columns derived from unique attributes.
- NOT NULL is specified on columns derived from required attributes.
- PRIMARY KEY is specified for primary key columns. The PRIMARY KEY keyword automatically enforces unique and required, so additional keywords NOT NULL and UNIQUE are unnecessary.

UNIQUE and NOT NULL are also specified on foreign key columns derived from unique and required relationships.

PARTICIPATION ACTIVITY 5.3.6: Implementing attribute cardinality.

Enter the missing keyword corresponding to the cardinality rules for the Student table below:

1)
```
CREATE TABLE Student (
  StudentNumber MEDIUMINT _____,
  CollegeName VARCHAR(20),
  EmailAddress VARCHAR(30)
)
```

primary key

Check Show answer

Correct

PRIMARY KEY

A solid bullet precedes StudentNumber, indicating StudentNumber is the primary key of the Student table. The PRIMARY KEY keyword also enforces unique and required attribute cardinality.

?????

Database design

The 'implementing attributes' step specifies columns, column constraints, and data types. Plural attributes become new weak tables. Unique and required attributes are implemented with UNIQUE, NOT NULL, and PRIMARY KEY keywords.

After the 'implementing attributes' step, the database is completely specified as CREATE TABLE statements. The final step, 'review tables for third normal form', ensures that tables do not contain redundant data and fine-tunes the design if necessary.

Table 5.3.1: Implement attributes.

Step	Activities
7A	Implement plural attributes as new weak tables.
7B	Specify cascade and restrict rules on new foreign keys in weak tables.
7C	Specify column data types corresponding to attribute types.
7D	Enforce relationship and attribute cardinality with UNIQUE and NOT NULL keywords.

TA time (Jordan) – 4 minutes
(CA 5.1.1 – Step 1 –Implementing strong entities)

CHALLENGE ACTIVITY | 5.1.1: Implementing entities.

Assume the following rules:

No two library cards have the same card number.
Each library card has a card number.
A card number is randomly assigned to each library card.
A library card's card number does not change.
A card number consists of six digits.

Which of the following properties does the CardNumber attribute have?

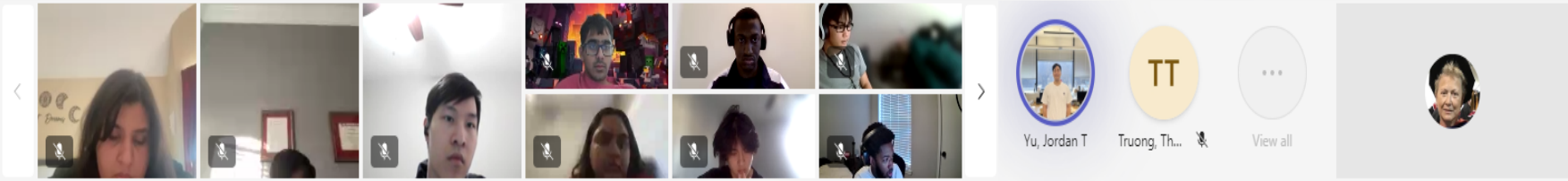
- [ ] Unique
- [ ] Required
- [ ] Stable
- [ ] Simple
- [ ] Meaningless

1

COSC 3380 23578 M & W 4 to 5:30 PM LECTURE (in TEAMS from 3:50 to 5:15 PM)

01:07:40

Take control | Pop out | Chat | People | Raise | View | Rooms | Apps | More | Camera | Mic | Share | Leave



Section 5.1 - COSC 3380: Da | Section 5.1 - COSC 3380: Da | Section 5.2 - COSC 3380: Da | AccessUH | Gradebook - 2024SP COSC3 | 2024SP COSC6397 23570 -

learn.zybooks.com/zybook/UHCOSC3380HilfordSpring2024/chapter/5/section/1

Inbox (1,324) | Gyazo - Free Scree... | Fantasy Basketball... | LeetCode - The W... | 12ft Ladder | COSC 6344 Visual... | AccessUH | Fantasy Basketball... | 2019_govyadinov_... | LintCode | All Bookmarks

zyBooks My library > COSC 3380: Database Systems home > 5.1: Implementing entities | zyBooks catalog | Help/FAQ | Jordan Yu

544874.877852.qx3zqy7

Jump to level 1

Assume the following rules:

A college name might include information like 'State', 'University', or 'College', as well as the academic area.
A college name can be long and hard to type.
A college may be renamed.
Every college has a name.
More than one college can have the same name.

Which of the following properties does the CollegeName attribute have?

- [ ] Unique
- [ ] Required
- [ ] Stable
- [ ] Simple
- [ ] Meaningless

1 | 2

Check | Next

✔ Expected: Required

CollegeName is:
Required. Every college must have a name.

However, CollegeName is:
Not unique. More than one college can have the same name.
Not stable. A college may be renamed.
Not simple. A college name can be too long to easily type in a WHERE clause.
Not meaningless. A college name might include descriptive information like 'State', 'University', or 'College', as well as the academic area.

View solution (Instructors only)

Feedback?

Yu, Jordan T

Participants

Invite someone or dial a number

Share invite

In this meeting (119) | Mute all

- Romero Ramirez, Gabr... 1
- Hilford, Victoria — Organizer
- Adhikari, Rohit
- Ahmed, Mohamed A
- Akram, Ali
- Akukwe, Benetta O
- Alsayed, Sami H
- Altaf, Sameer
- Alvarez, Stephanie
- Anayor-Achu, Ogochukwu E
- Avci, Hatice Kubra
- Aysola, Riya
- Bahl, Anish
- Banza, Sean Paolo B
- Bui, Hieu
- Burger, Jake

TA time (Jordan) – 4 minutes
(CA 5.1.1 – Step 2 –Implementing week entities)

CHALLENGE ACTIVITY | 5.1.1: Implementing entities.

Assume the following rules:

Track meets have events with names such as "Men's 4x100m relay".
Each event has a short, three-letter code.
The primary key of Meet is MeetID.

Select the best primary key for the Event table.

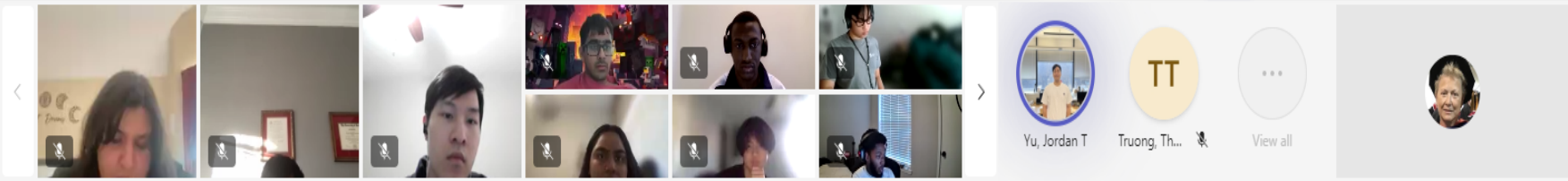
Select
- Select
- (MeetID, EventCode)
- (MeetID, EventName)
- EventName
- MeetID

2

COSC 3380 23578 M & W 4 to 5:30 PM LECTURE (in TEAMS from 3:50 to 5:15 PM)

01:09:31

Take control | Pop out | Chat | People | Raise | View | Rooms | Apps | More | Camera | Mic | Share | Leave



Yu, Jordan T | Truong, Th... | View all

Section 5.1 - COSC 3380: Da | Section 5.1 - COSC 3380: Da | Section 5.2 - COSC 3380: Da | AccessUH | Gradebook - 2024SP COSC3 | 2024SP COSC6397 23570 -

learn.zybooks.com/zybook/UHCOSC3380HilfordSpring2024/chapter/5/section/1

Inbox (1,324) | Gyazo - Free Scree... | Fantasy Basketball... | LeetCode - The W... | 12ft Ladder | COSC 6344 Visual... | AccessUH | Fantasy Basketball... | 2019_govyadinov_... | LintCode | All Bookmarks

zyBooks My library > COSC 3380: Database Systems home > 5.1: Implementing entities | zyBooks catalog | Help/FAQ | Jordan Yu

ACTIVITY 5.1.1: Implementing entities

544874.877952.qx3zqy7

Jump to level 1

Assume the following rules:

An orchard can take orders from many wholesalers, and a wholesaler can submit orders to many orchards.
A shipping company has a database with tables Orchard, Order, and Wholesaler.
The primary key of Orchard is OrchardID, and the primary key of Wholesaler is WholesalerID.
A wholesaler can place only one order per day with a particular orchard.

Select the best primary key for the Order table.

(OrchardID, WholesalerID, OrderDate)

1 | 2

Check | Next **Done**. Click any level to practice more. Completion is preserved.

✔ Expected: (OrchardID, WholesalerID, OrderDate)

Order is identified by both Orchard and Wholesaler. So, the primary key of Order must be a composite that includes the primary keys of Orchard and Wholesaler.
However, a particular wholesaler may place orders with a particular orchard on many different dates.
Therefore, OrderDate must be included to make the key unique.

View solution (Instructors only)

Feedback?

How was this section? Provide section feedback

Yu, Jordan T

Participants

Invite someone or dial a number

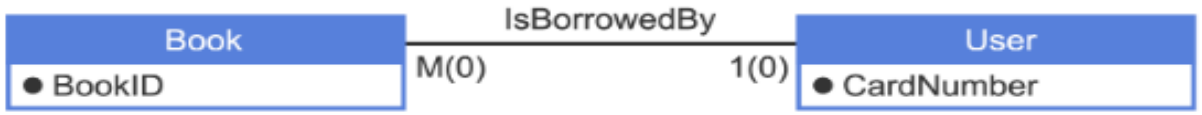
Share invite

In this meeting (119) | Mute all

- Romero Ramirez, Gabr... 1
- Hilford, Victoria — Organizer
- Adhikari, Rohit
- Ahmed, Mohamed A
- Akram, Ali
- Akukwe, Benetta O
- Alsayed, Sami H
- Altaf, Sameer
- Alvarez, Stephanie
- Anayor-Achu, Ogochukwu E
- Avci, Hatice Kubra
- Aysola, Riya
- Bahl, Anish
- Banza, Sean Paolo B
- Bui, Hieu
- Burger, Jake

TA time (Jordan) – 4 minutes
(CA 5.2.1 – Step 1 –Implementing relationships)

CHALLENGE ACTIVITY | 5.2.1: Implementing relationships.



In which table is the foreign key placed?

Pick

What is the name of the new foreign key?

Pick

Are NULLs allowed in the foreign key column?

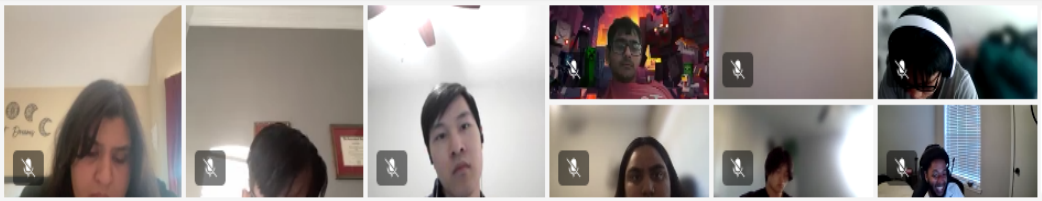
Pick

1

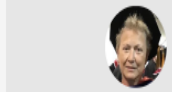
COSC 3380 23578 M & W 4 to 5:30 PM LECTURE (in TEAMS from 3:50 to 5:15 PM)

01:12:06

Take control | Pop out | Chat | People | Raise | View | Rooms | Apps | More | Camera | Mic | Share | Leave



Yu, Jordan T | Truong, Th... | View all



Section 5.1 - COSC 3380: D | Section 5.1 - COSC 3380: D | Section 5.2 - COSC 3380: D | AccessUH | Gradebook - 2024SP COSC | 2024SP COSC6397 23570 -

learn.zybooks.com/zybook/UHCOSC3380HilfordSpring2024/chapter/5/section/2

Inbox (1,324) | Gyazo - Free Scree... | Fantasy Basketball... | LeetCode - The W... | 12ft Ladder | COSC 6344 Visual... | AccessUH | Fantasy Basketball... | 2019_govyadinov_... | LintCode | All Bookmarks

zyBooks My library > COSC 3380: Database Systems home > 5.2: Implementing relationships

zyBooks catalog | Help/FAQ | Jordan Yu

These entities and attributes become tables and primary keys.

Implement the relationship as a foreign key.

In which table is the foreign key placed?

Campus

What is the name of the new foreign key?

UniversityID

Are NULLs allowed in the foreign key column?

No

1 2 3 4

Check | Next

✔ Expected: Campus, UniversityID, No

The foreign key goes in the Campus table, the 'many' side of the relationship.
The foreign key refers to UniversityID, the primary key on the 'one' side.
NULLs are not allowed in the UniversityID column of the Campus table, because University is required in the relationship.

View solution (Instructors only)

Feedback?

Yu, Jordan T

Participants

Invite someone or dial a number

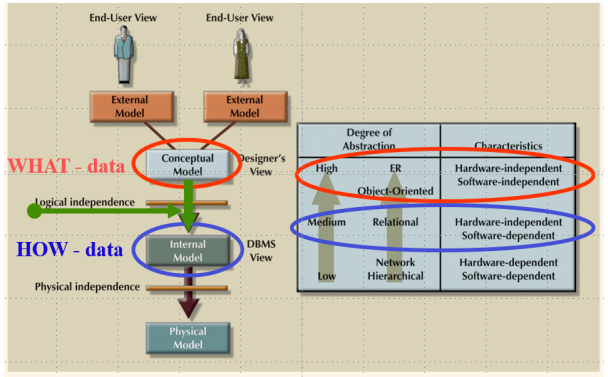
Share invite

In this meeting (118) | Mute all

- GR Romero Ramirez, Gabr... 1
- Hilford, Victoria — Organizer
- RA Adhikari, Rohit
- MA Ahmed, Mohamed A
- AA Akram, Ali
- BA Akukwe, Benetta O
- SA Alsayed, Sami H
- SA Altaf, Sameer
- SA Alvarez, Stephanie
- OA Anayor-Achu, Ogochukwu E
- HA Avci, Hatice Kubra
- RA Aysola, Riya
- AB Bahl, Anish
- SB Banza, Sean Paolo B
- HB Bui, Hieu
- Burger, Jake

From 5:00 to 5:05 PM – 5 minutes

01.29.2024 (4 - Mo)	ZyBook SET 1 - 4	Set 1 LECTURE 4 ERD to RELATIONAL



WHAT HOW

1. SET 1 Empty

5. SET 1 - 4:ERD to RELATIONAL 0% 0%

Please work on

SET 1 – 5: ERD to RELATIONAL MODEL

SET 1 Lecture 4
ERD to RELATIONAL

5. SET 1 - 4:ERD to RELATIONAL

- 5.1 Implementing entities Hidden
- 5.2 Implementing relationships Hidden
- 5.3 Implementing attributes Hidden

Next

01.31.2024 (5 - We)	ZyBook SET 1 - 5	Set 1 LECTURE 5 NORMALIZATION

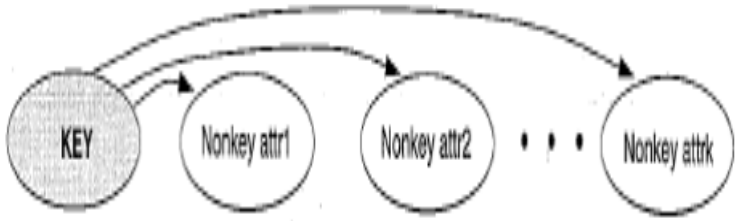
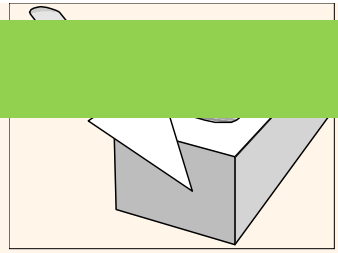


Figure 19.5 FDs in a BCNF Relation

1. SET 1 Empty

6. SET 1 - 5:NORMALIZATION 0% 0%

From 5:05 to 5:15 PM – 5 minutes.



01.29.2024 (4 - Mo)	ZyBook SET 1 - 4	Set 1 LECTURE 4 ERD to RELATIONAL

CLASS PARTICIPATION 20 points — 20% of Total

CLASS 4

Class 4 END PARTICIPATION

Not available until Jan 29 at 5:05pm | Due Jan 29 at 5:15pm | 100 pts

This is a synchronous online class.

Attendance is required.

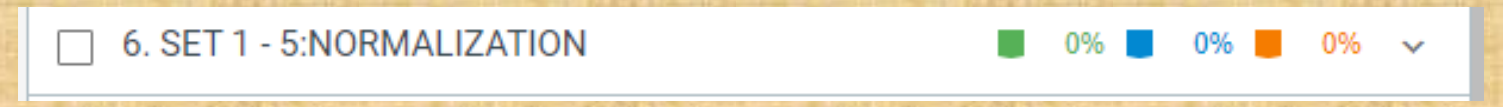
Recording or distribution of class materials is prohibited.

1. At the beginning of selected classes there is an **assessment in the first 10 minutes**. (**beige** BOX in the Detailed Syllabus)
2. At the end of selected classes there is an **assessment in the last 10 minutes**. (**blue** BOX in the Detailed Syllabus)
3. ZyBook sections will be downloaded and used for 30% of Total Score on the dates specified in the Detailed Syllabus.
4. EXAMS are in CANVAS. **No late EXAMS**.
5. I have to be **present in TEAMS in order to take any graded assignment assigned during that class.**

At 5:15 PM.

End Class 4

VH, unhide ZyBook Section 6.



VH, Download Attendance Report
Rename it:
1.29.2024 Attendance Report FINAL

VH, upload Class 4 to CANVAS.