The Relational Data Model

Grigorios Loukides
Email: grigorios.loukides@kcl.ac.uk

Session Objectives

In this session, you will learn:

- Terminology of the relational model
- How tables are used to represent data
- Properties of database relations
- Meaning of entity integrity and referential integrity.
- How to use EntityRelationship (ER) modelling in database design.
- Basic concepts associated with ER model.
- Diagrammatic technique for displaying ER model
- How to build an ER model from a requirements specification.

Database Design Problem

Imaging you want to store information about a martial arts club.

The club has a number of members who can practice different martial arts. For all members you need to register, in addition to the name and the membership number, which martial arts they practice (you can practice several) and which belt (or degree) they hold in the art in question. You also need to register information about their membership fee, namely the amount and the payment date. You must even register whether the member has a valid licence or not (you need a different licence for each art that you practice, so think about how to represent these!)

1. How we can create a database to store this information?

Database Design Problem

The club has a number of members who can practice different martial arts. For all members you need to register, in addition to the name and the membership number, which martial arts they practice (you can practice several) and which belt (or degree) they hold in the art in question. You also need to register information about their membership fee, namely the amount and the payment date. You must even register whether the member has a valid licence or not (you need a different licence for each art that you practice, so think about how to represent these!)

1. How we can create a database to store this information?
    1. Which database model?
    2. How to define the elements of the model?
    3. How to describe them?

Database Design Problem

The club has a number of members who can practice different martial arts. For all members you need to register, in addition to the name and the membership number, which martial arts they practice (you can practice several) and which belt (or degree) they hold in the art in question. You also need to register information about their membership fee, namely the amount and the payment date. You must even register whether the member has a valid licence or not (you need a different licence for each art that you practice, so think about how to represent these!)

1. How we can create a database to store this information?
    1. Which database model?
    2. How to define the elements of the model?
    3. How to describe them?

Introduction

- The **relational model** was introduced by E. F. Codd[1]
- Relational databases have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personnel data, ...

[1] IBM's San Jose Research Laboratory

Introduction

- Relational model
  - View data logically rather than physically
- Logical view of relational database is based on relation
  - Relation thought of as a table
- Table: two-dimensional structure composed of rows and columns
  - Contains group of related entities (entity set)

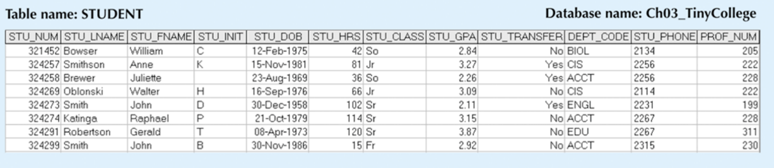
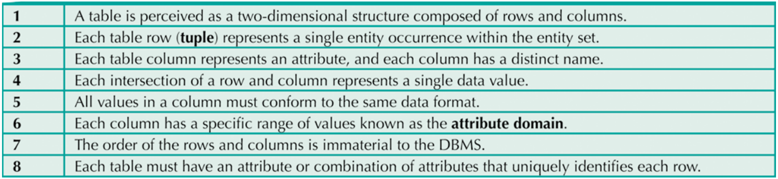


Table name: STUDENT **Database name: Ch03_TinyCollege**

STU_NUM	STU_LNAME	STU_FNAME	STU_INIT	STU_DOB	STU_HRS	STU_CLASS	STU_GPA	STU_TRANSFER	DEPT_CODE	STU_PHONE	PROF_NUM
321452	Bowser	William	C	12-Feb-1975	42	So	2.84	No	BIOL	2134	205
324257	Smithson	Anne	K	15-Nov-1981	81	Jr	3.27	Yes	CIS	2256	222
324258	Brewer	Juliette		23-Aug-1969	36	So	2.26	Yes	ACCT	2256	228
324269	Oblonski	Walter	H	16-Sep-1976	66	Jr	3.09	No	CIS	2114	222
324273	Smith	John	D	30-Dec-1958	102	Sr	2.11	Yes	ENGL	2231	199
324274	Katinga	Raphael	P	21-Oct-1979	114	Sr	3.15	No	ACCT	2267	228
324291	Robertson	Gerald	T	08-Apr-1973	120	Sr	3.87	No	EDU	2267	311
324299	Smith	John	B	30-Nov-1986	15	Fr	2.92	No	ACCT	2315	230

Characteristics of a Relational Table



1	A table is perceived as a two-dimensional structure composed of rows and columns.
2	Each table row (**tuple**) represents a single entity occurrence within the entity set.
3	Each table column represents an attribute, and each column has a distinct name.
4	Each intersection of a row and column represents a single data value.
5	All values in a column must conform to the same data format.
6	Each column has a specific range of values known as the **attribute domain**.
7	The order of the rows and columns is immaterial to the DBMS.
8	Each table must have an attribute or combination of attributes that uniquely identifies each row.

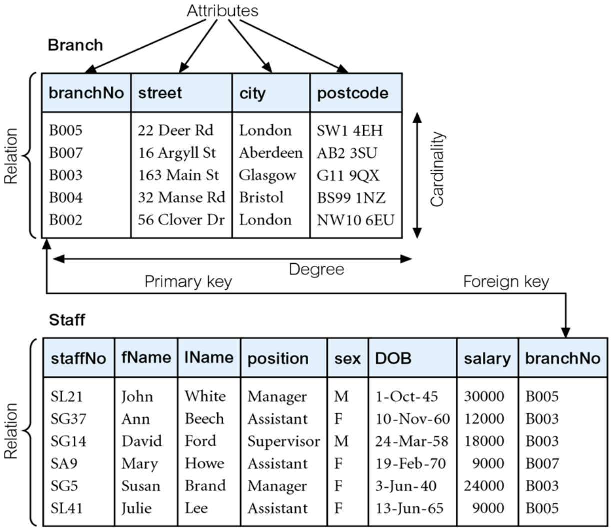
Concepts

- Relation: Is a table with columns and rows
- Tuple: Is a row of the relation
- Cardinality: Is the number of tuples in a relation
- Attribute: Is a named column of a relation
- Domain: Is the set of allowable values for one or more attributes
- Degree: Is the number of attributes in a relation

Relational Database

Is a collection of normalized relations with distinct relation names

Concepts



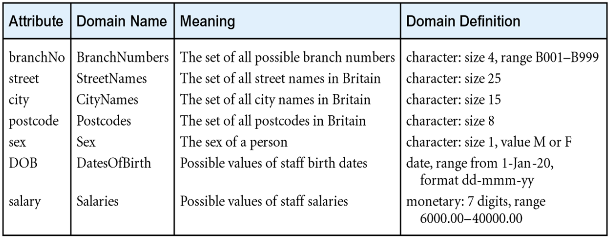
Branch

branchNo	street	city	postcode
B005	22 Deer Rd	London	SW1 4EH
B007	16 Argyll St	Aberdeen	AB2 3SU
B003	163 Main St	Glasgow	G11 9QX
B004	32 Manse Rd	Bristol	BS99 1NZ
B002	56 Clover Dr	London	NW10 6EU

Staff

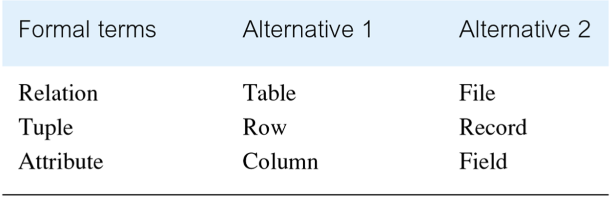
staffNo	fName	lName	position	sex	DOB	salary	branchNo
SL21	John	White	Manager	M	1-Oct-45	30000	B005
SG37	Ann	Beech	Assistant	F	10-Nov-60	12000	B003
SG14	David	Ford	Supervisor	M	24-Mar-58	18000	B003
SA9	Mary	Howe	Assistant	F	19-Feb-70	9000	B007
SG5	Susan	Brand	Manager	F	3-Jun-40	24000	B003
SL41	Julie	Lee	Assistant	F	13-Jun-65	9000	B005

Concepts



Attribute	Domain Name	Meaning	Domain Definition
branchNo	BranchNumbers	The set of all possible branch numbers	character: size 4, range B001–B999
street	StreetNames	The set of all street names in Britain	character: size 25
city	CityNames	The set of all city names in Britain	character: size 15
postcode	Postcodes	The set of all postcodes in Britain	character: size 8
sex	Sex	The sex of a person	character: size 1, value M or F
DOB	DatesOfBirth	Possible values of staff birth dates	date, range from 1-Jan-20, format dd-mmm-yy
salary	Salaries	Possible values of staff salaries	monetary: 7 digits, range 6000.00–40000.00

Concepts



Formal terms	Alternative 1	Alternative 2
Relation	Table	File
Tuple	Row	Record
Attribute	Column	Field

Database Relations

Relation schema

Named relation defined by a set of attribute and domain name pairs

Relational database schema

Set of relation schemas, each with a distinct name

Properties

- Relation name is distinct from all other relation names in relational schema
- Each cell of relation contains exactly one atomic (single) value
- Each attribute has a distinct name
- Values of an attribute are all from the same domain
- Each tuple is distinct; there are no duplicate tuples
- Order of attributes has no significance
- Order of tuples has no significance, theoretically

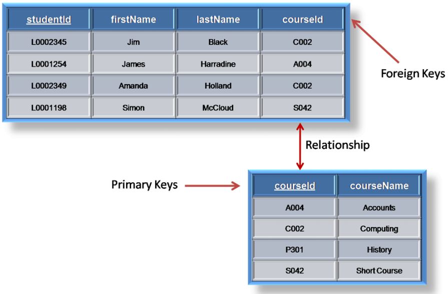
Keys

Primary Key

An attribute, or set of attributes, selected to identify tuples uniquely within relation

Foreign Key

Attribute, or set of attributes, within one relation that matches primary key of some (possibly same) relation



studentId	firstName	lastName	courseId
L0002345	Jim	Black	C002
L0001254	James	Harradine	A004
L0002349	Amanda	Holland	C002
L0001198	Simon	McCloud	S042

courseId	courseName
A004	Accounts
C002	Computing
P301	History
S042	Short Course

Example

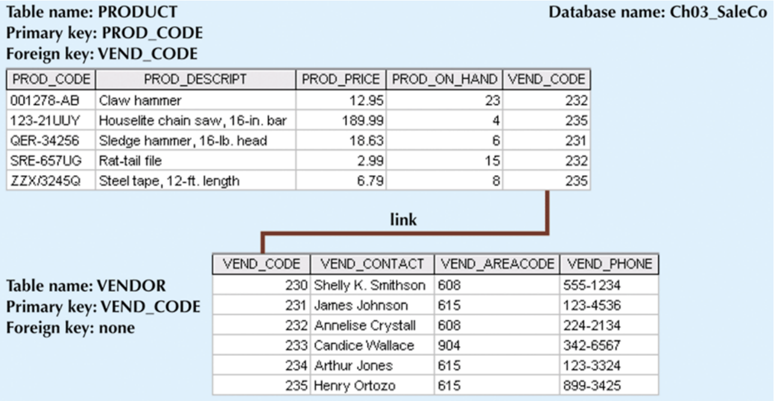


Table name: PRODUCT
Primary key: PROD_CODE
Foreign key: VEND_CODE

Database name: Ch03_SaleCo

PROD_CODE	PROD_DESCRIPT	PROD_PRICE	PROD_ON_HAND	VEND_CODE
001278-AB	Claw hammer	12.95	23	232
123-21UUY	Houselite chain saw, 16-in. bar	189.99	4	235
QER-34256	Sledge hammer, 16-lb. head	18.63	6	231
SRE-657UG	Rat-tail file	2.99	15	232
ZZX/3245Q	Steel tape, 12-ft. length	6.79	8	235

Table name: VENDOR
Primary key: VEND_CODE
Foreign key: none

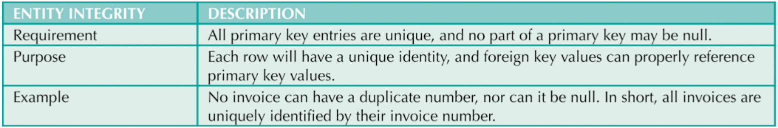
VEND_CODE	VEND_CONTACT	VEND_AREACODE	VEND_PHONE
230	Shelly K. Smithson	608	555-1234
231	James Johnson	615	123-4536
232	Annelise Crystall	608	224-2134
233	Candice Wallace	904	342-6567
234	Arthur Jones	615	123-3324
235	Henry Ortozo	615	899-3425

Integrity Constraints

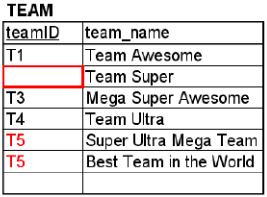
Integrity constraints are used to ensure accuracy and consistency of data in a relational database:

- Entity Integrity
- Referential Integrity
- General Constraints

Entity Integrity



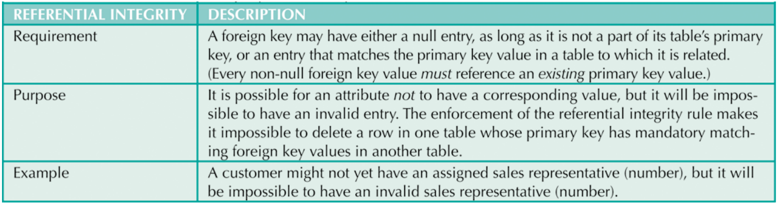
ENTITY INTEGRITY	DESCRIPTION
Requirement	All primary key entries are unique, and no part of a primary key may be null.
Purpose	Each row will have a unique identity, and foreign key values can properly reference primary key values.
Example	No invoice can have a duplicate number, nor can it be null. In short, all invoices are uniquely identified by their invoice number.



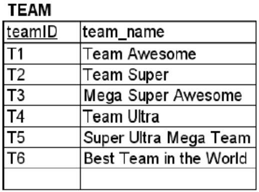
TEAM

teamID	team_name
T1	Team Awesome
	Team Super
T3	Mega Super Awesome
T4	Team Ultra
T5	Super Ultra Mega Team
T5	Best Team in the World

Referential Integrity

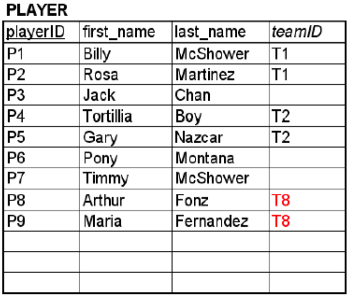


REFERENTIAL INTEGRITY	DESCRIPTION
Requirement	A foreign key may have either a null entry, as long as it is not a part of its table's primary key, or an entry that matches the primary key value in a table to which it is related. (Every non-null foreign key value *must* reference an *existing* primary key value.)
Purpose	It is possible for an attribute *not* to have a corresponding value, but it will be impossible to have an invalid entry. The enforcement of the referential integrity rule makes it impossible to delete a row in one table whose primary key has mandatory matching foreign key values in another table.
Example	A customer might not yet have an assigned sales representative (number), but it will be impossible to have an invalid sales representative (number).



TEAM

teamID	team_name
T1	Team Awesome
T2	Team Super
T3	Mega Super Awesome
T4	Team Ultra
T5	Super Ultra Mega Team
T6	Best Team in the World



PLAYER

playerID	first_name	last_name	*teamID*
P1	Billy	McShower	T1
P2	Rosa	Martinez	T1
P3	Jack	Chan	
P4	Tortillia	Boy	T2
P5	Gary	Nazcar	T2
P6	Pony	Montana	
P7	Timmy	McShower	
P8	Arthur	Fonz	T8
P9	Maria	Fernandez	T8

General Constraints

Additional rules specified by users or database administrators that define or constrain some aspect of the enterprise

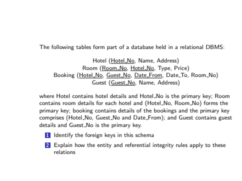
Exercise

The following tables form part of a database held in a relational DBMS:

Hotel (Hotel_No, Name, Address)
Room (Room_No, Hotel_No, Type, Price)
Booking (Hotel_No, Guest_No, Date_From, Date_To, Room_No)
Guest (Guest_No, Name, Address)

where Hotel contains hotel details and Hotel_No is the primary key; Room contains room details for each hotel and (Hotel_No, Room_No) forms the primary key; booking contains details of the bookings and the primary key comprises (Hotel_No, Guest_No and Date_From); and Guest contains guest details and Guest_No is the primary key.

1. Identify the foreign keys in this schema
2. Explain how the entity and referential integrity rules apply to these relations



The following tables form part of a database held in a relational DBMS:

Hotel (Hotel_No, Name, Address)
Room (Room_No, Hotel_No, Type, Price)
Booking (Hotel_No, Guest_No, Date_From, Date_To, Room_No)
Guest (Guest_No, Name, Address)

where Hotel contains hotel details and Hotel_No is the primary key; Room contains room details for each hotel and (Hotel_No, Room_No) forms the primary key; booking contains details of the bookings and the primary key comprises (Hotel_No, Guest_No and Date_From); and Guest contains guest details and Guest_No is the primary key.

1. Identify the foreign keys in this schema
2. Explain how the entity and referential integrity rules apply to these relations

1. Identify the foreign keys in this schema:
   Hotel_No in Room, Hotel_No in Booking, Guest_No in Booking
2. Explain how the entity and referential integrity rules apply to these relations:
   Not null and unique:Hotel_No in Hotel, (Room_No, Hotel_No) in Room, (Hotel_No, Guest_No, Date_From) in Booking, Guest_Noin Guest
   Existing in the reference table (all FK)

Database Design Problem

The club has a number of members who can practice different martial arts. For all members you need to register, in addition to the name and the membership number, which martial arts they practice (you can practice several) and which belt (or degree) they hold in the art in question. You also need to register information about their membership fee, namely the amount and the payment date. You must even register whether the member has a valid licence or not (you need a different licence for each art that you practice, so think about how to represent these!)

1. How we can create a database to store this information?
    1. Which database model? Relational Model
    2. How to define the elements of the model?
    3. How to represent them?

Entity Relationship Model

The Entity Relationship Model:

1. Is a conceptual design for a relational database
2. Helps to understand and to specify the desired components of the database and the relationships among those components
3. Entity Relationship Model represents conceptual database as viewed by end user

Entity Relationship Model: Building Blocks

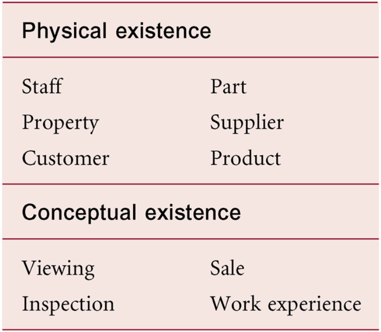
ER has 3 main components:

- Entity: anything about which data are to be collected and stored
    - Eg: STUDENT, COURSE are entities
- Attribute: a characteristic of an entity
    - Eg: For a STUDENT entity name, surname, DOB are attributes
- Relationship: describes an association among entities
    - Eg: STUDENT takes COURSE

Entity

- Entities represent objects or things of interest:
  - Physical things like students, lecturers, employees, products
  - More abstract things like modules, orders, courses, projects
- Entities have
  - A general type or class, such as Lecturer or Module
  - Instances of that particular type, such as Natalia is an instance of Lecturer
  - Attributes (such as name, email address)

Examples of Entities



Physical existence	
Staff	Part
Property	Supplier
Customer	Product
Conceptual existence	
Viewing	Sale
Inspection	Work experience

Attribute

- Attribute
  - Property of an entity
- Attribute Domain
  - Set of allowable values for one or more attributes.

Relationships

- Relationships represent associations among entity types
- Main types:
  - One-to-many (1:M) relationship
    - Eg: STUDENT submits an ASSESMENT
  - Many-to-many (M:N or M:M) relationship
    - Eg: STUDENT takes COURSE
  - One-to-one (1:1) relationship
    - Eg: STUDENT manages KCL_ACCOUNT

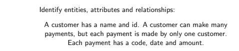
Translating Requirements into ER Components

- Some nouns translate into entities
- Some nouns translate into attributes
- Some verbs translate into relationships among entities

Example

Identify entities, attributes and relationships:

> A customer has a name and id. A customer can make many payments, but each payment is made by only one customer. Each payment has a code, date and amount.



Identify entities, attributes and relationships:

A customer has a name and id. A customer can make many payments, but each payment is made by only one customer. Each payment has a code, date and amount.

Identify entities, attributes and relationships:

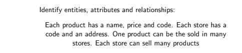
A customer has a name and id. A customer can make many payments, but each payment is made by only one customer. Each payment has a code, date and amount.

- Entities: customer, payment
- Attributes:
  - Customer: name,id
  - Payment: code, date, amount
- Relationship: makes 1:N (One to many)

Exercise

Identify entities, attributes and relationships:

> Each product has a name, price and code. Each store has a code and an address. One product can be the sold in many stores. Each store can sell many products



Identify entities, attributes and relationships:

Each product has a name, price and code. Each store has a code and an address. One product can be the sold in many stores. Each store can sell many products

Identify entities, attributes and relationships:

Each product has a name, price and code. Each store has a code and an address. One product is sold in many stores. Each store sells many products

- Entities: product, store
- Attributes:
  - Product: name , price, code
  - Store: code address
- Relationship: sells M:N (many to many)

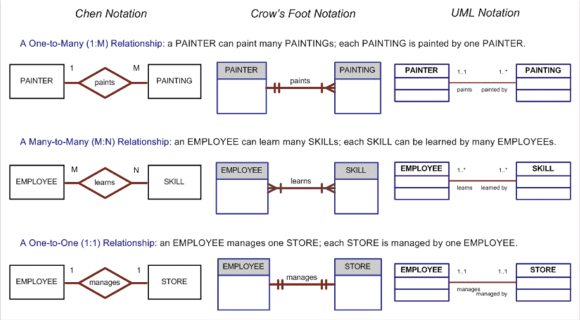
Database Design Problem

The club has a number of members who can practice different martial arts. For all members you need to register, in addition to the name and the membership number, which martial arts they practice (you can practice several) and which belt (or degree) they hold in the art in question. You also need to register information about their membership fee, namely the amount and the payment date. You must even register whether the member has a valid licence or not (you need a different licence for each art that you practice, so think about how to represent these!)

1. How we can create a database to store this information?
    1. Which database model? Relational Model
    2. How to define the elements of the model? ER Model
    3. How to represent them?

Entity Relationship Diagram

- Uses graphic representations represent the Entity Relationship Model



A One-to-Many (1:M) Relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER.

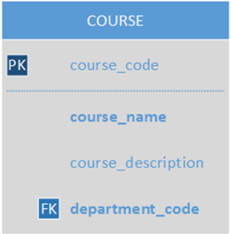
A Many-to-Many (M:N) Relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs.

A One-to-One (1:1) Relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE.

- No one set of symbols dominates industry use, and none is necessarily better than another

Crow's Foot Notation: Entities

- Entity is usually drawn as a box
- The box is labelled with the name of the class of objects represented by that entity
- Attributes are represented inside the box



COURSE

PK course_code

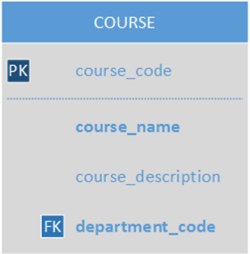
course_name

course_description

FK department_code

Crow's Foot Notation: Attributes

- Attributes have a name and a data type
- Different symbols are used to represent the different types of attributes



COURSE

PK course_code

course_name

course_description

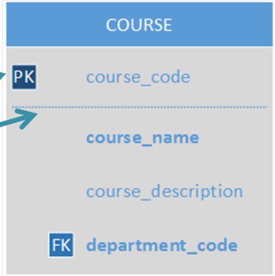
FK department_code

Crow's Foot Notation: Attributes I

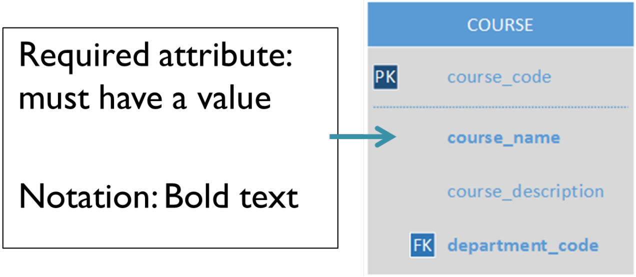
Primary Key Notation:

PK symbol (optional)

Line



Crow's Foot Notation: Attributes II



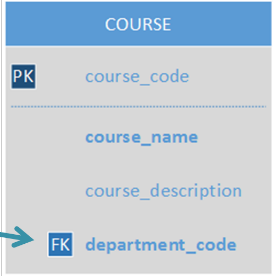
Required attribute: must have a value

Notation: Bold text

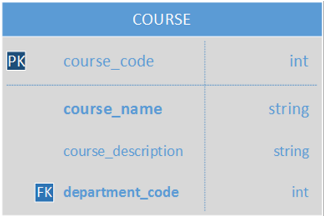
Crow's Foot Notation: Attributes III

Foreign Key Notation:

FK symbol



Crow’s Foot Notation: Attributes IV



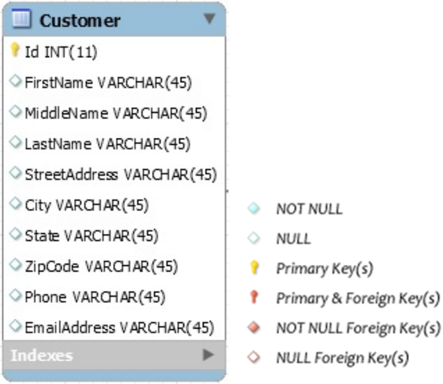
COURSE

PK	course_code	int
	course_name	string
	course_description	string
FK	**department_code**	int

- INT
- FLOAT
- STRING
- DATE
- TIME
- DATETIME

Crow's Foot Notation: Attributes V

MySQL uses different symbols:



Customer

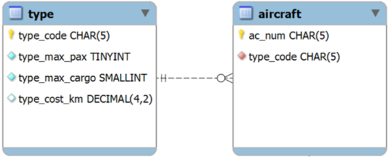
- Id INT(11)
- ◇ FirstName VARCHAR(45)
- ◇ MiddleName VARCHAR(45)
- ◇ LastName VARCHAR(45)
- ◇ StreetAddress VARCHAR(45)
- ◇ City VARCHAR(45)
- ◇ State VARCHAR(45)
- ◇ ZipCode VARCHAR(45)
- ◇ Phone VARCHAR(45)
- ◇ EmailAddress VARCHAR(45)

Indexes ▶

- ◆ *NOT NULL*
- ◇ *NULL*
- *Primary Key(s)*
- *Primary & Foreign Key(s)*
- ◆ *NOT NULL Foreign Key(s)*
- ◇ *NULL Foreign Key(s)*

Crow's Foot Notation: Relationships

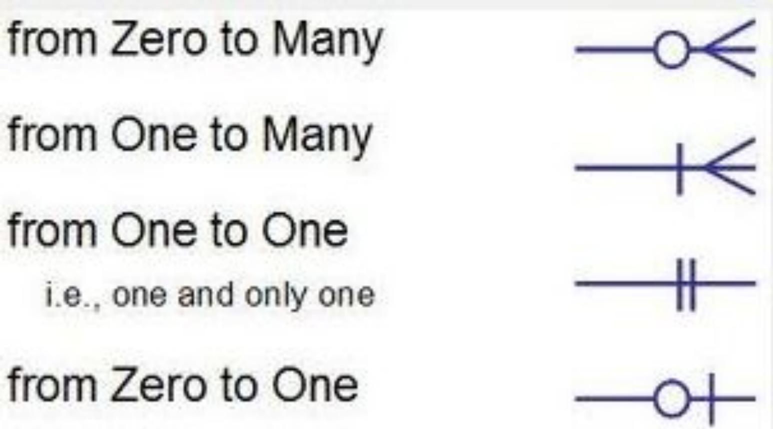
- Relationships are represented as links between two entities
- The name may be given on top of the link (more than one relationships between entities)
- The ends of the link show cardinality



type
type_code CHAR(5)
type_max_pax TINYINT
type_max_cargo SMALLINT
type_cost_km DECIMAL(4,2)

aircraft
ac_num CHAR(5)
type_code CHAR(5)

Crow’s Foot Notation: Relationships II



from Zero to Many

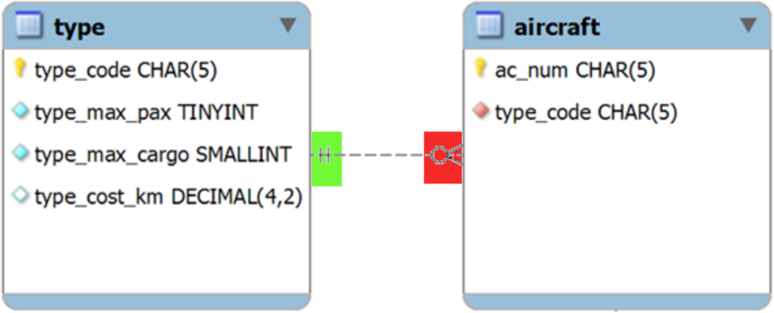
from One to Many

from One to One

i.e., one and only one

from Zero to One

Understanding Relationships in ERD



type

- type_code CHAR(5)
- type_max_pax TINYINT
- type_max_cargo SMALLINT
- type_cost_km DECIMAL(4,2)

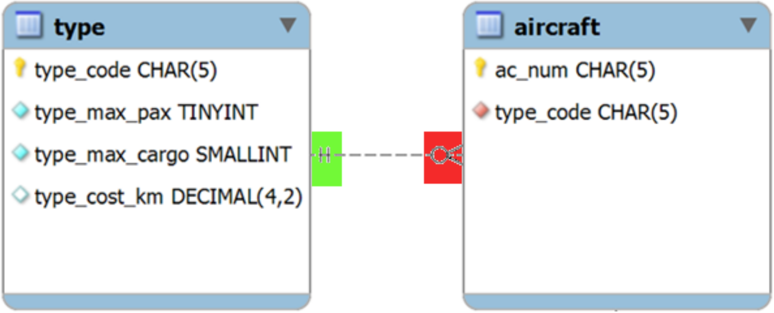
aircraft

- ac_num CHAR(5)
- type_code CHAR(5)

How many aircrafts are related to one type?

How many types are related to one aircraft?

Understanding Relationships in ERD



type

- type_code CHAR(5)
- type_max_pax TINYINT
- type_max_cargo SMALLINT
- type_cost_km DECIMAL(4,2)

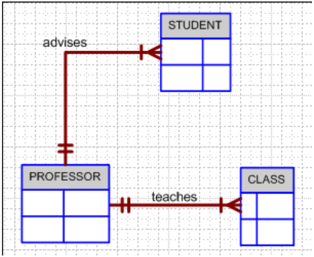
aircraft

- ac_num CHAR(5)
- type_code CHAR(5)

How many aircrafts are related to one type? 0 or more

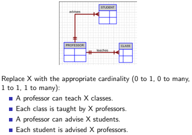
How many types are related to one aircraft? One and just one

Exercise



Replace X with the appropriate cardinality (0 to 1, 0 to many, 1 to 1, 1 to many):

- A professor can teach X classes.
- Each class is taught by X professors.
- A professor can advise X students.
- Each student is advised X professors.



Replace X with the appropriate cardinality (0 to 1, 0 to many, 1 to 1, 1 to many):

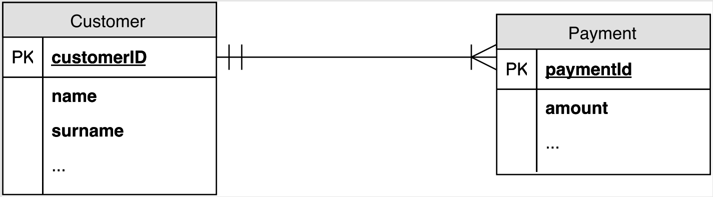
- A professor can teach X classes.
- Each class is taught by X professors.
- A professor can advise X students.
- Each student is advised X professors.

- A professor can teach 1 to MANY classes.
- Each class is taught by ONE professors.
- A professor can advise 1 to MANY students.
- Each student is advised ONE professor.

Exercise

A customer can make many payments, but each payment is made by only one customer

A customer can make many payments, but each payment is made by only one customer



Customer

PK	**customerID**
	name **surname** ...

Payment

PK	**paymentId**
	amount ...

Implementing Relationships

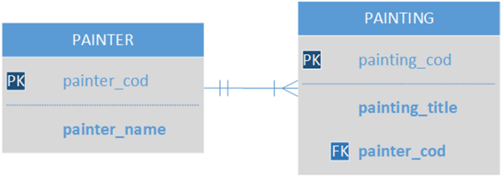
1:M relationship
: Relational modelling ideal
: Should be the norm in any relational database design

1:1 relationship
: Should be rare in any relational database design

M:N relationships
: Cannot be implemented as such in the relational model
: M:N relationships can be changed into 1:M relationships

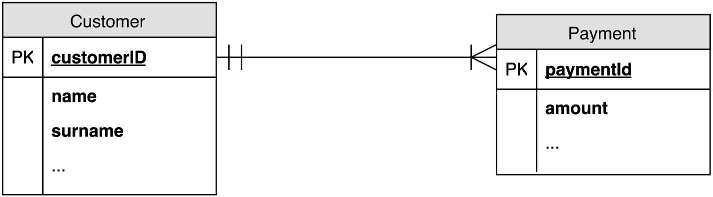
The 1:M Relationship

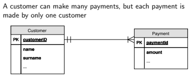
- It is implemented by a FK
- Some software may draw the FK when you create the relationship. If not you should add it to your model



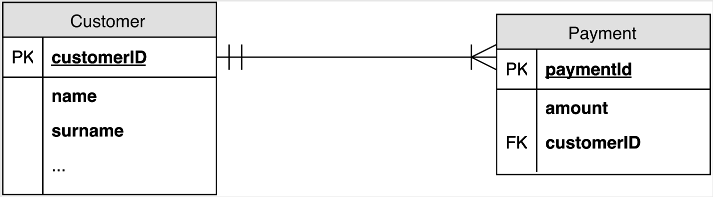
Exercise

A customer can make many payments, but each payment is made by only one customer



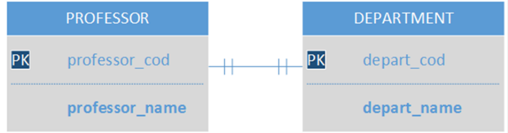


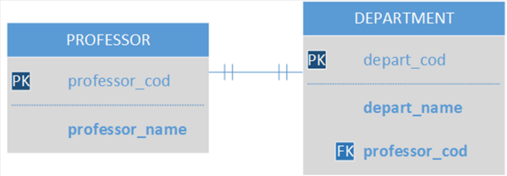
A customer can make many payments, but each payment is made by only one customer



The 1:1 Relationship

- Sometimes means that entity components were not defined properly
- If one side optional, implemented as a FK in the optional side
- If not, implemented as a FK on any side





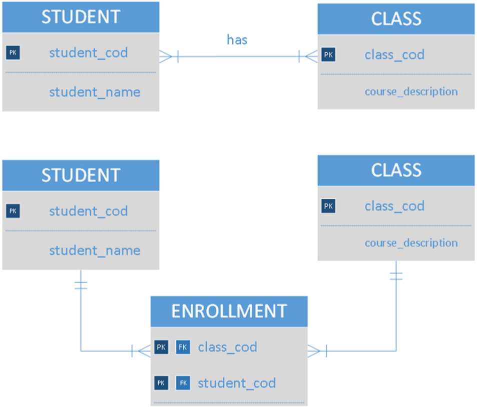
The N:M Relationship

- Implemented by breaking it up to produce a set of 1:M relationships
- Avoid problems inherent to M:N relationship by creating a composite entity
- Includes as foreign keys the primary keys of tables to be linked

Composite entities

- Also known as bridge entities
- Used to implement M:N relationships
- Composed of primary keys of each of the entities to be connected
- May also contain additional attributes that play no role in connective process

Converting the M:N Relationship into two 1:N relationships

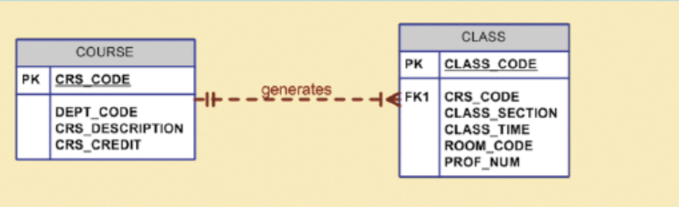


Relationship Strength

- Weak (non-identifying) relationships (dashed line)
  - Exists if PK of related entity does not contain PK component of parent entity
- Strong (identifying) relationships
  - Exists when PK of related entity contains PK component of parent entity

Relationship Strength

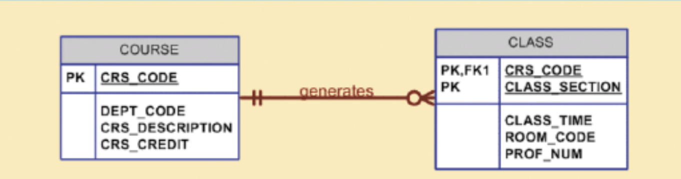
A weak (non-identifying) relationship between COURSE and CLASS



COURSE	
PK	CRS_CODE
	DEPT_CODE CRS_DESCRIPTION CRS_CREDIT

CLASS	
PK	CLASS_CODE
FK1	CRS_CODE CLASS_SECTION CLASS_TIME ROOM_CODE PROF_NUM

A strong (identifying) relationship between COURSE and CLASS



COURSE	
PK	CRS_CODE
	DEPT_CODE CRS_DESCRIPTION CRS_CREDIT

CLASS	
PK,FK1 PK	CRS_CODE CLASS_SECTION
	CLASS_TIME ROOM_CODE PROF_NUM

Drawing ER Models

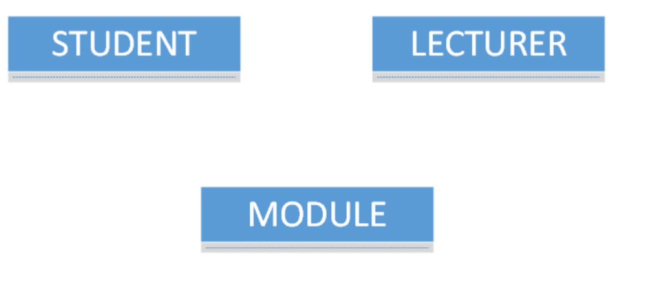
Database design is an iterative process

1. Identify main entities and relationships from business rules
2. Develop initial ERD
3. Identify attributes and primary keys that adequately describe entities
4. Implement relationships (Add FK, bridge entities, etc)
5. Identify weak and strong relationships
6. Revise and review ERD

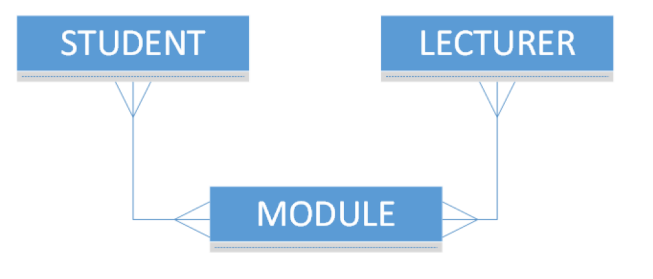
Exercise

KCL has asked you to design a database to record details of lecturers who deliver modules and the students who attend their modules. For each module, KCL wants to know the code, name, semester and year. The id, name, and speciality of each lecturer must be recorded. For each student, record details of his id, name and date of birth

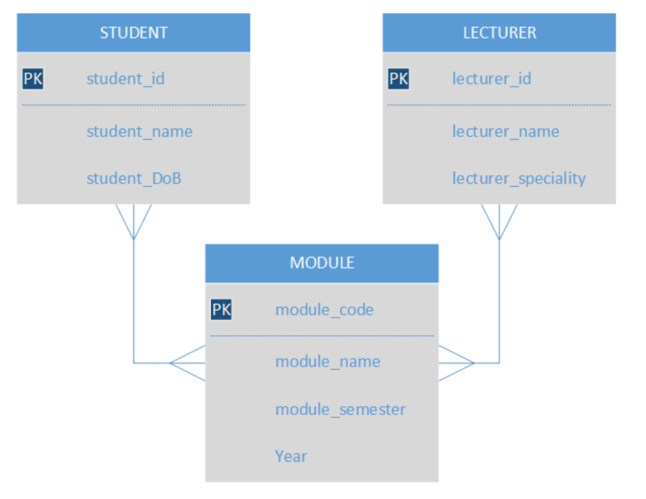
KCL has asked you to design a database to record details of lecturers who deliver modules and the students who attend their modules. For each module, KCL wants to know the code, name, semester and year. The id, name, and speciality of each lecturer must be recorded. For each student, record details of his id, name and date of birth



KCL has asked you to design a database to record details of lecturers who deliver modules and the students who attend their modules. For each module, KCL wants to know the code, name, semester and year. The id, name, and speciality of each lecturer must be recorded. For each student, record details of his id, name and date of birth



KCL has asked you to design a database to record details of lecturers who deliver modules and the students who attend their modules. For each module, KCL wants to know the code, name, semester and year. The id, name, and speciality of each lecturer must be recorded. For each student, record details of his id, name and date of birth



STUDENT

PK student_id

student_name

student_DoB

LECTURER

PK lecturer_id

lecturer_name

lecturer_speciality

MODULE

PK module_code

module_name

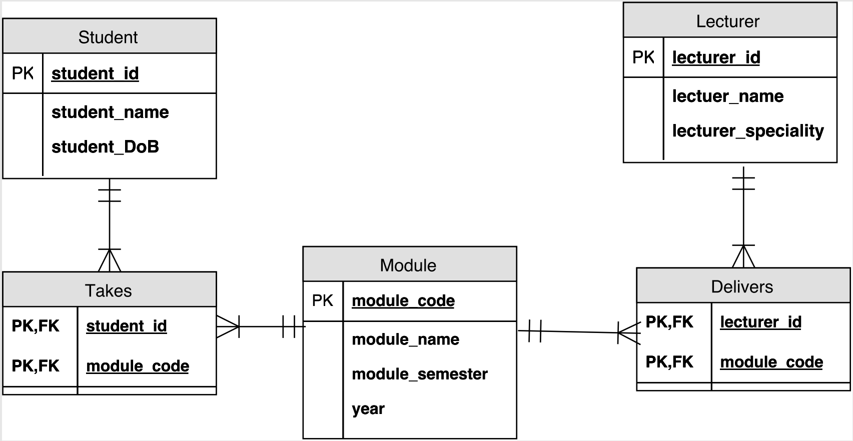
module_semester

Year

2018-09-26

The Relational Data Model
Entity Relationship Diagram
Drawing ER Models
Exercise

KCL has asked you to design a database to record details of lecturers who deliver modules and the students who attend their modules. For each module, KCL wants to know the code, name, semester and year. The id, name, and speciality of each lecturer must be recorded. For each student, record details of his id, name and date of birth



Student

PK **student_id**

student_name

student_DoB

Takes

PK,FK **student_id**

PK,FK **module_code**

Module

PK **module_code**

module_name

module_semester

year

Lecturer

PK **lecturer_id**

lectuer_name

lecturer_speciality

Delivers

PK,FK **lecturer_id**

PK,FK **module_code**

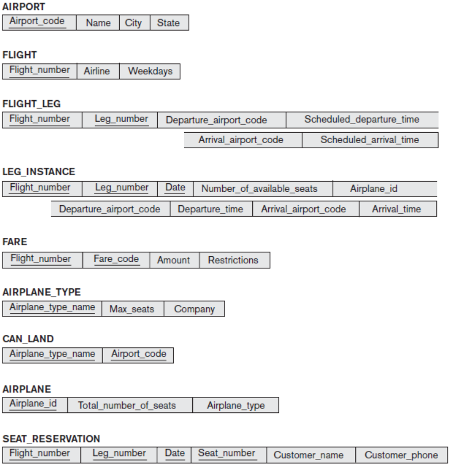
Database Design Problem

Draw an ERD for next week:

The club has a number of members who can practice different martial arts. For all members you need to register, in addition to the name and the membership number, which martial arts they practice (you can practice several) and which belt (or degree) they hold in the art in question. You also need to register information about their membership fee, namely the amount and the payment date. You must even register whether the member has a valid licence or not (you need a different licence for each art that you practice, so think about how to represent these!)

Exercise 1

Specify all the referential and integrity constraints that hold on this schema



AIRPORT

Airport_code	Name	City	State

FLIGHT

Flight_number	Airline	Weekdays

FLIGHT_LEG

Flight_number	Leg_number	Departure_airport_code	Scheduled_departure_time	Arrival_airport_code	Scheduled_arrival_time

LEG_INSTANCE

Flight_number	Leg_number	Date	Number_of_available_seats	Airplane_id	Departure_airport_code	Departure_time	Arrival_airport_code	Arrival_time

FARE

Flight_number	Fare_code	Amount	Restrictions

AIRPLANE_TYPE

Airplane_type_name	Max_seats	Company

CAN_LAND

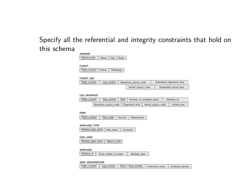
Airplane_type_name	Airport_code

AIRPLANE

Airplane_id	Total_number_of_seats	Airplane_type

SEAT_RESERVATION

Flight_number	Leg_number	Date	Seat_number	Customer_name	Customer_phone



Specify all the referential and integrity constraints that hold on this schema

AIRPORT

Airport_code	Name	City	State

FLIGHT

Flight_number	Airline	Weekdays

FLIGHT_LEG

Flight_number	Leg_number	Departure_airport_code	Scheduled_departure_time	Arrival_airport_code	Scheduled_arrival_time

LEG_INSTANCE

Flight_number	Leg_number	Date	Number_of_available_seats	Airplane_id	Departure_airport_code	Departure_time	Arrival_airport_code	Arrival_time

FARE

Flight_number	Fare_code	Amount	Restrictions

AIRPLANE_TYPE

Airplane_type_name	Max_seats	Company

CAN_LAND

Airplane_type_name	Airport_code

AIRPLANE

Airplane_id	Total_number_of_seats	Airplane_type

SEAT_RESERVATION

Flight_number	Leg_number	Date	Seat_number	Customer_name	Customer_phone

Table: Flight_Leg
Entity Integrity: Each combination of flight_number and leg_numer value is unique and there are no nulls.
Refernatial Integrity: Each flight_number value in flight_leg points to an existing flight_number value in flight.

....

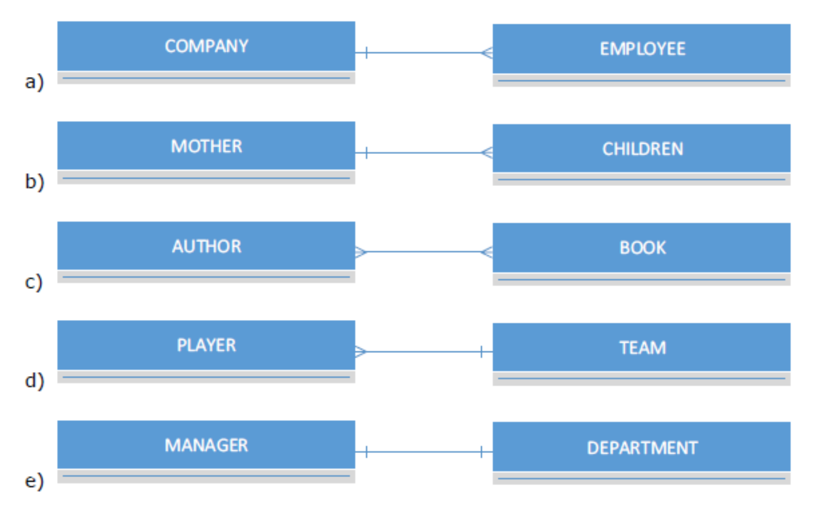
Exercise 2

Draw ER diagrams for the following Business Rules, showing Entities, Relationships and Cardinality.

- A department has many employees. An employee works in only one department.
- A mother may have many children. A child has one mother.
- An author can write many books. A book may be written by many authors.
- A player plays for only one team. A team consists of many players.
- A manager manages at most one department. A department is managed by at most one manager.

Draw ER diagrams for the following Business Rules, showing Entities, Relationships and Cardinality.

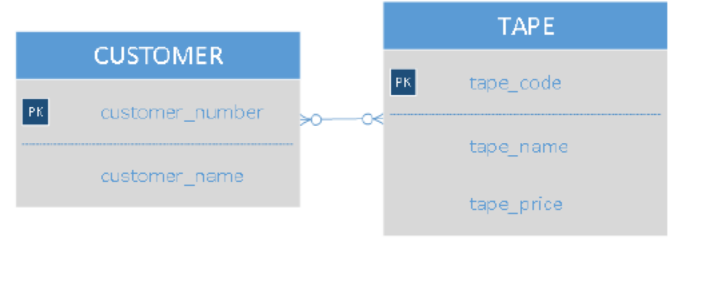
- A department has many employees. An employee works in only one department.
- A mother may have many children. A child has one mother.
- An author can write many books. A book may be written by many authors.
- A player plays for only one team. A team consists of many players.
- A manager manages at most one department. A department is managed by at most one manager.



Exercise 3

Customers have a number and a name. Tapes have a name, a code and a price. A Customer may rent many tapes. A tape can be rented by many customers

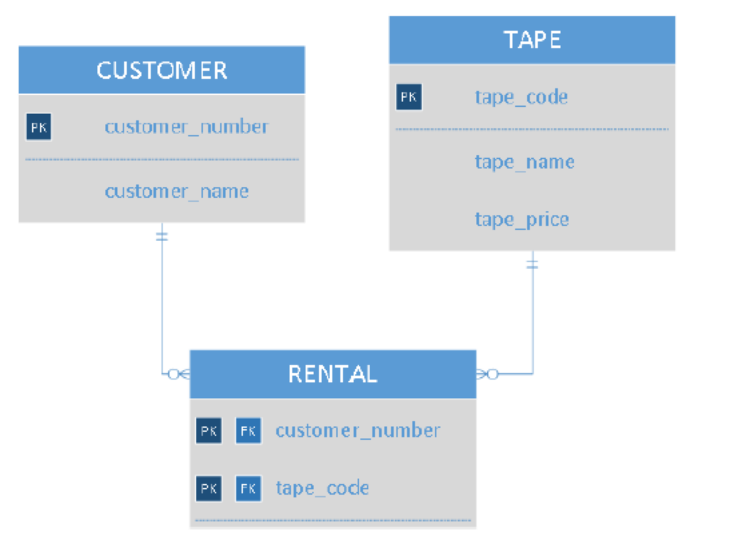
Customers have a number and a name. Tapes have a name, a code and a price. A Customer may rent many tapes. A tape can be rented by many customers



2018-09-26

The Relational Data Model
└─Tutorial Exercises
└─Exercise 3

Customers have a number and a name. Tapes have a name, a code and a price. A Customer may rent many tapes. A tape can be rented by many customers



Exercise 4

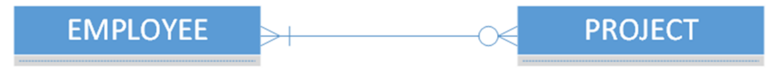
A company has a number of employees that can be identified by Employee_ID. Each employee has a name, address and date of birth. The company also has several projects, which can be identified by Project number. The project details given to the company are project name and project start date. Each employee may be assigned to one or more project, or may not be assigned to a project. A project must have at least one more employee assigned, and may have any number of employees assigned

Exercise 4

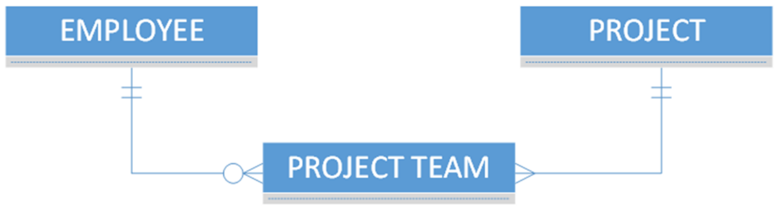
A company has a number of employees that can be identified by Employee_ID. Each employee has a name, address and date of birth. The company also has several projects, which can be identified by Project number. The project details given to the company are project name and project start date. Each employee may be assigned to one or more project, or may not be assigned to a project. A project must have at least one more employee assigned, and may have any number of employees assigned

EMPLOYEE
PROJECT

A company has a number of employees that can be identified by Employee_ID. Each employee has a name, address and date of birth. The company also has several projects, which can be identified by Project number. The project details given to the company are project name and project start date. Each employee may be assigned to one or more project, or may not be assigned to a project. A project must have at least one more employee assigned, and may have any number of employees assigned



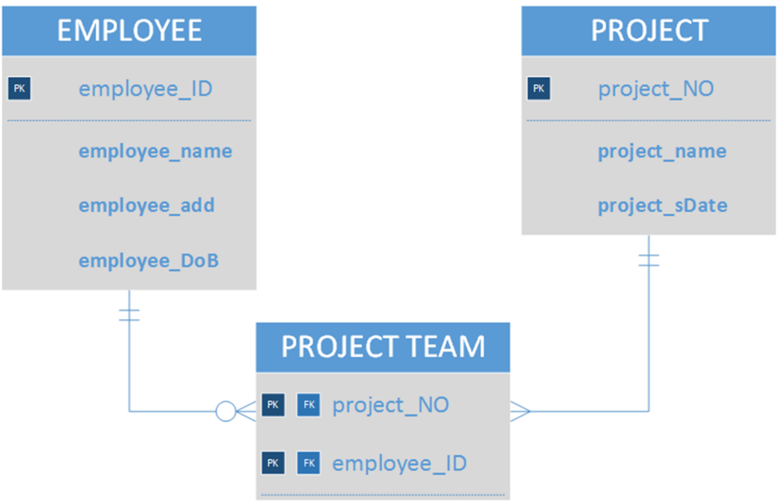
A company has a number of employees that can be identified by Employee_ID. Each employee has a name, address and date of birth. The company also has several projects, which can be identified by Project number. The project details given to the company are project name and project start date. Each employee may be assigned to one or more project, or may not be assigned to a project. A project must have at least one more employee assigned, and may have any number of employees assigned



2018-09-26

The Relational Data Model
└Tutorial Exercises
└Exercise 4

A company has a number of employees that can be identified by Employee_ID. Each employee has a name, address and date of birth. The company also has several projects, which can be identified by Project number. The project details given to the company are project name and project start date. Each employee may be assigned to one or more project, or may not be assigned to a project. A project must have at least one more employee assigned, and may have any number of employees assigned



EMPLOYEE
PK employee_ID
employee_name
employee_add
employee_DoB

PROJECT
PK project_NO
project_name
project_sDate

PROJECT TEAM
PK FK project_NO
PK FK employee_ID

Conclusion

In this session we have covered:

- The relational model
  - Concepts, properties, keys and integrity constraints
- The Entity Relationship Model
  - Entity Relationship Diagram (Crow's foot notation)

Lab Session

Next week lab session is about accessing MySQL from Python and drawing ERD

Suggested Readings

- Chapters 3 and 7 of Fundamentals of Database Systems. Elmasri & Navathe.
- Chapters 3 and 11 of Database systems: a practical approach to design, implementation, and management. Connolly, Thomas M; Begg, Carolyn

Drawing Tools

- Microsoft Visio, OneNote, Powerpoint, Illustrator
- You can also use web apps such as http://draw.io to help you sketch out your diagrams.