INFO20003: Database Systems

Dr Renata Borovica-Gajic

Lecture 23

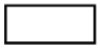
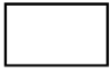
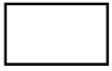
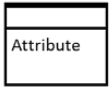
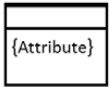
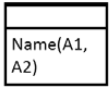
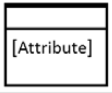
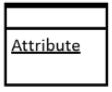
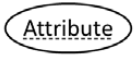
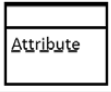
Overview, sample exam questions

Part I

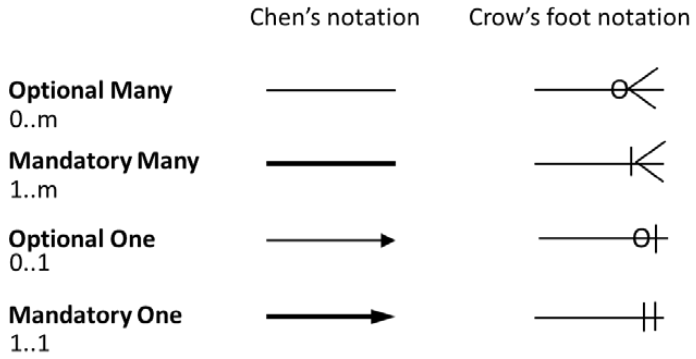
Week 12

Overview

Modelling – learn notations

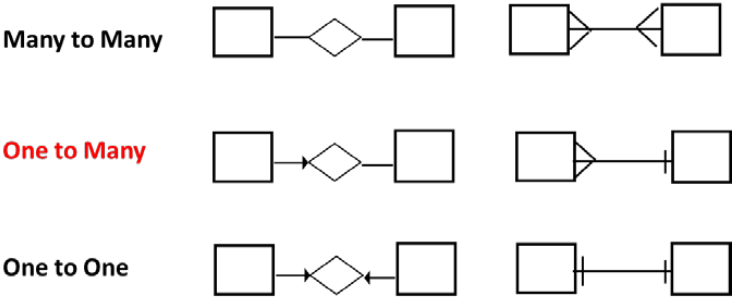
Concept	Chen's not.	Crow's foot not.	
Entity			
Weak Entity			
Attribute			
Multivalued A.			*
Composite A.			*
Derived A.			*
Key A.			*
Weak (or Partial) Key A.			*
Relationship			
Weak (Identify-ing) Relationship			

Relationship cardinalities and constraints



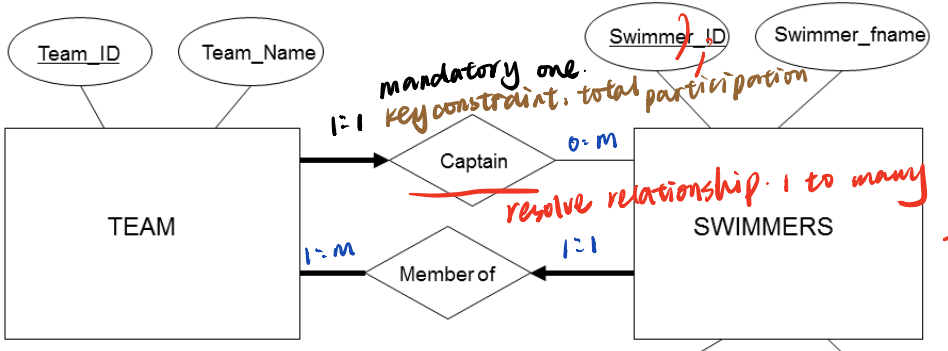
BINARY Relationship Cardinalities

Here we just looked at cardinalities and omitted participation constraints (optional/mandatory) for clarity



Example

Write SQL statements to create the tables for the data model below. Be sure to specify primary and foreign keys. You do not need to specify whether the fields are NULL/NOT NULL.



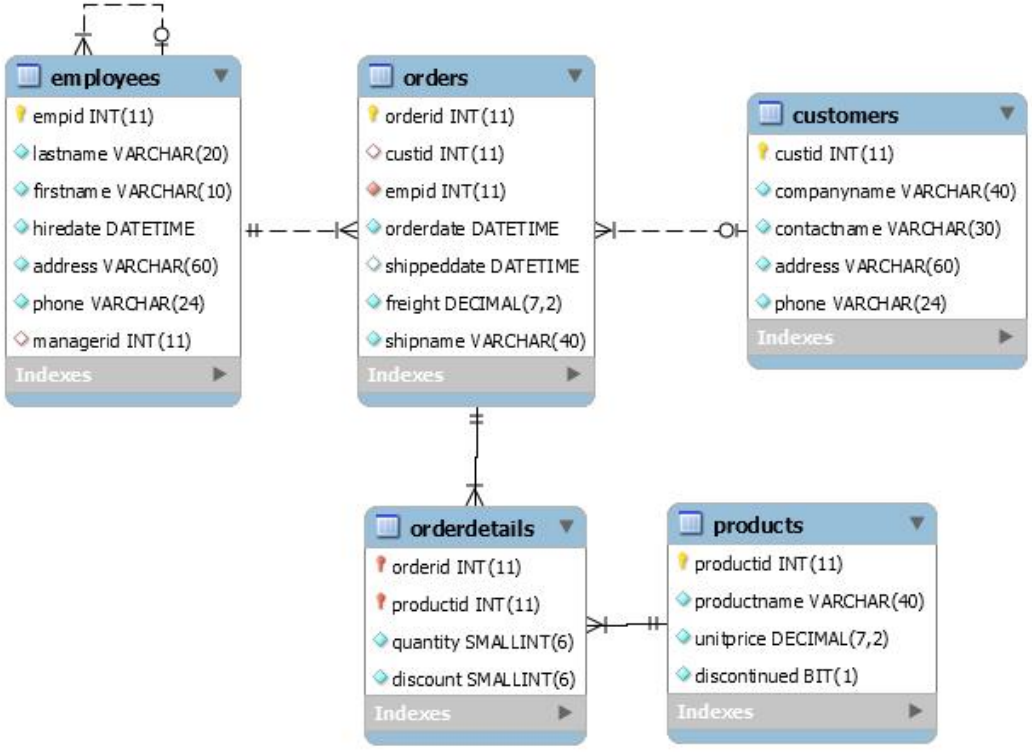
CREATE TABLE TEAM (
Team_ID INT,
Team_Name VARCHAR(20),
Captain_ID INT,
PRIMARY KEY (TeamID),
foreign key (captain_ID) references Swimmers (swimmer_ID)
);

(Chen's notation: move many side to one side).

PARTIAL KEY
WEAK ENTITY

CREATE TABLE SWIMMERS (
SwimmerID INT,
SwimmerFname VARCHAR(25),
SwimmerLname VARCHAR(25),
SwimmerEmail VARCHAR(100),
TEAM-ID INT,
PRIMARY KEY (SwimmerID),
FOREIGN KEY (Team-ID) REFENCE TEAM (TEAM_ID)
);

Given the following schema, write a *single* SQL statement to correctly answer each of the following questions. DO NOT USE VIEWS to answer questions.

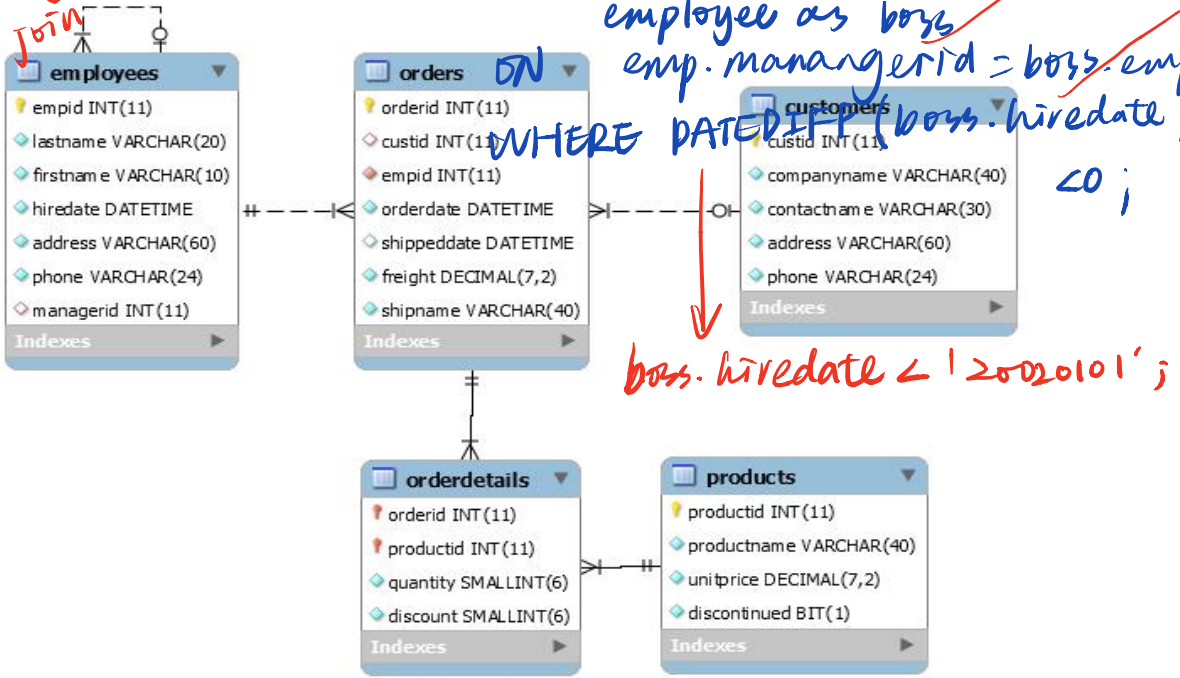


SQL: Q1

Write a query that returns the first name and last name of employees whose manager was hired prior to 01/01/2002.

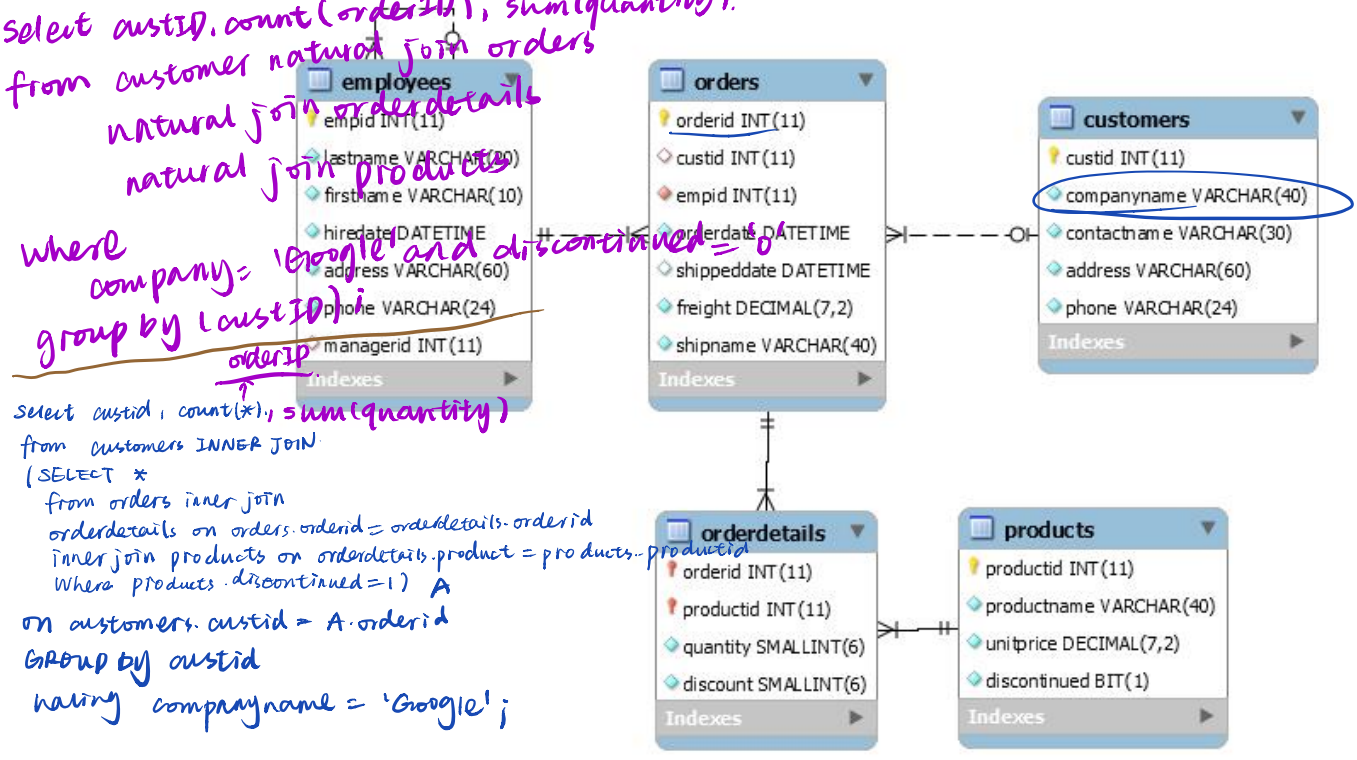
```
Select emp.firstname, emp.lastname
From employee as emp INNER JOIN
     employee as boss
     ON emp.managerid = boss.empid
WHERE DATEDIFF(boss.hiredate, 01/01/2002) < 0;

boss.hiredate < '20020101';
```

alias to distinguish for many join



SQL: Q2

Write a query that returns customers IDs whose company name is 'Google', and for each customer return the total number of orders and total quantities for all products that were not discontinued ('1' means discontinued, '0' not discontinued).




```
select custID, count(orderID), sum(quantity)
from customer natural join orders
natural join orderdetails
natural join products
where company = 'Google' and discontinued = 0
group by (custID);
```

```
select custid, count(*), sum(quantity)
from customers INNER JOIN
(SELECT *
  from orders inner join
  orderdetails on orders.orderid = orderdetails.orderid
  inner join products on orderdetails.product = products.productid
  Where products.discontinued = 1) A
on customers.custid = A.orderid
GROUP by custid
having companyname = 'Google';
```

Next lecture

- Sample exam questions Part II