COMP9120

Week 13: Review

Semester 2, 2022

THE UNIVERSITY OF SYDNEY

Acknowledgement of Country

I would like to acknowledge the Traditional Owners of Australia and recognise their continuing connection to land, water and culture. I am currently on the land of the Gadigal people of the Eora nation and pay my respects to their Elders, past, present and emerging.

I further acknowledge the Traditional Owners of the country on which you are on and pay respects to their Elders, past, present and future.

Copyright warning

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by or on behalf of the University of Sydney pursuant to Part VB of the Copyright Act 1968 (**the Act**).

The material in this communication may be subject to copyright under the Act. Any further copying or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice.

Guest lecturer

Muhammad Aamir Cheema

- Associate Professor, Monash University
- ARC Future Fellow

Recognition:

- 2012 Malcolm Chaikin Prize for Research Excellence in Engineering
- 2013 Discovery Early Career Researcher Award
- 2014 Dean's Award for Excellence in Research by an Early Career Researcher
- 2018 Future Fellowship, 2018 Monash Student Association Teaching Award and
- 2019 Young Tall Poppy Science Award.

Research Focus

› Urban Computing: a process of creation, acquisition, integration, and analysis of big and heterogeneous data generated by diverse sources in urban spaces, such as sensors, devices, vehicles, buildings, humans and other species, to tackle major issues that cities face (e.g., air pollution, increased energy consumption, and traffic congestion).

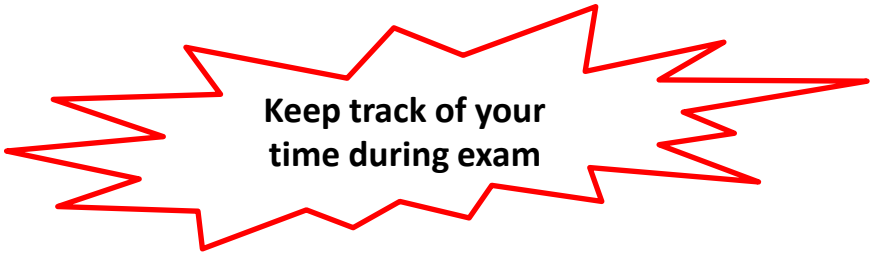
Profile page: http://www.aamircheema.com/

THE UNIVERSITY OF SYDNEY

5-min break!

Final Exam

- Date: Wednesday 16th November
- Time: 1:00 PM Sydney time
- Duration: 175 Minutes
  - Reading time: 10 Minutes
  - Writing time: 150 Minutes
  - Upload time: 15 minutes

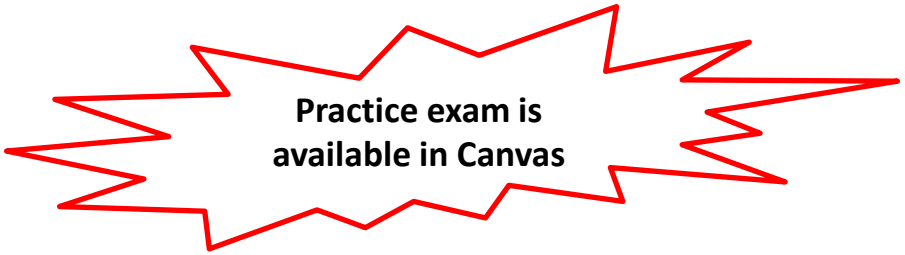


- Everyone starts the exam at the same time
  - No late submission
- Exam adjustment is done by the exam office
  - Notification no later than 3 days before the exam

Final Exam

Question Type

- No MCQs
- Essay-type questions (10 Questions→50 marks)
  - ERD
  - Relational Model
  - Relational Algebra
  - SQL
  - Integrity Constraints
  - Transactions
  - Normalizations
  - Storage
  - Indexing
  - Query Processing



› Exam covers 50% of the final mark

› You must obtain at least **40% in the exam**, as well as an **overall mark of at least 50%**, to pass the unit

Course Outline

	Week	Topic
Foundations	Week 1	Introduction
	Week 2	Conceptual Database Design
	Week 3	Relational Data Model / Logical Database Design
	Week 4	Relational Algebra and SQL
	Week 5	Complex SQL
	Week 6	Database Integrity
Applications	~~Week 7~~	~~Database Application Development and Security~~
	Week 8	Transaction Management
	Week 9	Schema Refinement and Normalisation
Internals	Week 10	Storage and Indexing
	Week 11	Query Evaluation and Optimisation
	Week 12	Revision
	Week 13	Final exam Structure

Which is the correct model?

"Each chef prepares **at most** one dish. Every dish must be prepared by **at least** one chef."

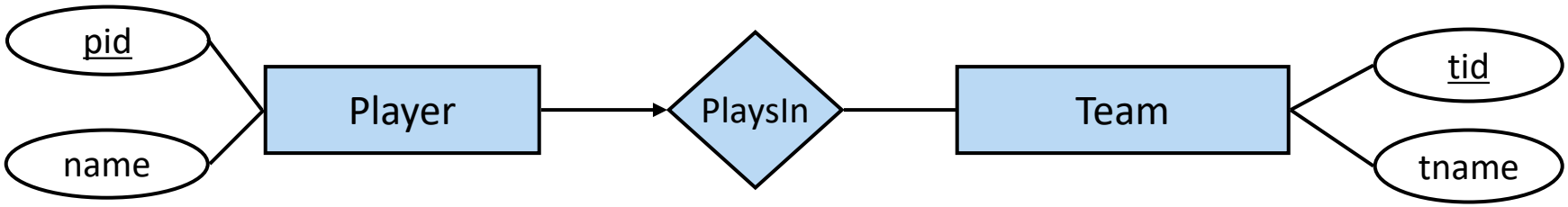
a. 

b. 

c. 

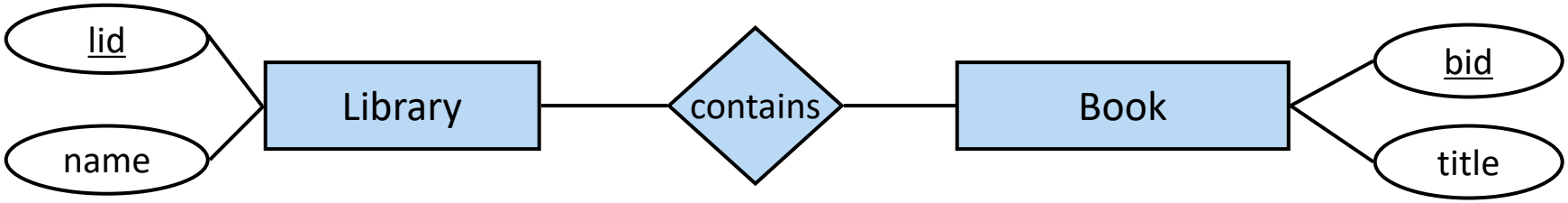
d. 

Mapping Relationship Types with Key Constraints

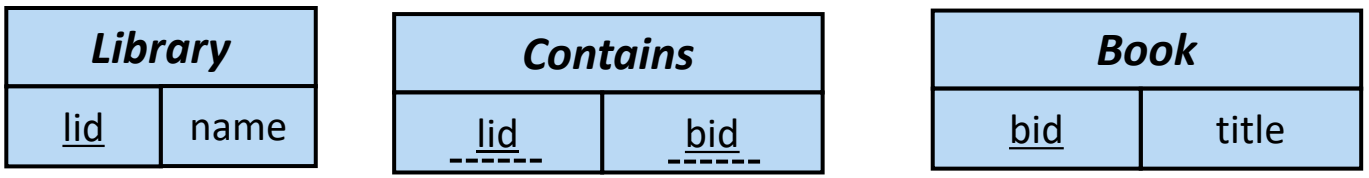


- Looking on each relationship side:
  - 1 player plays in at most 1 team
  - 1 team can have 0 to Many players

Mapping Relationship Types with Key Constraints

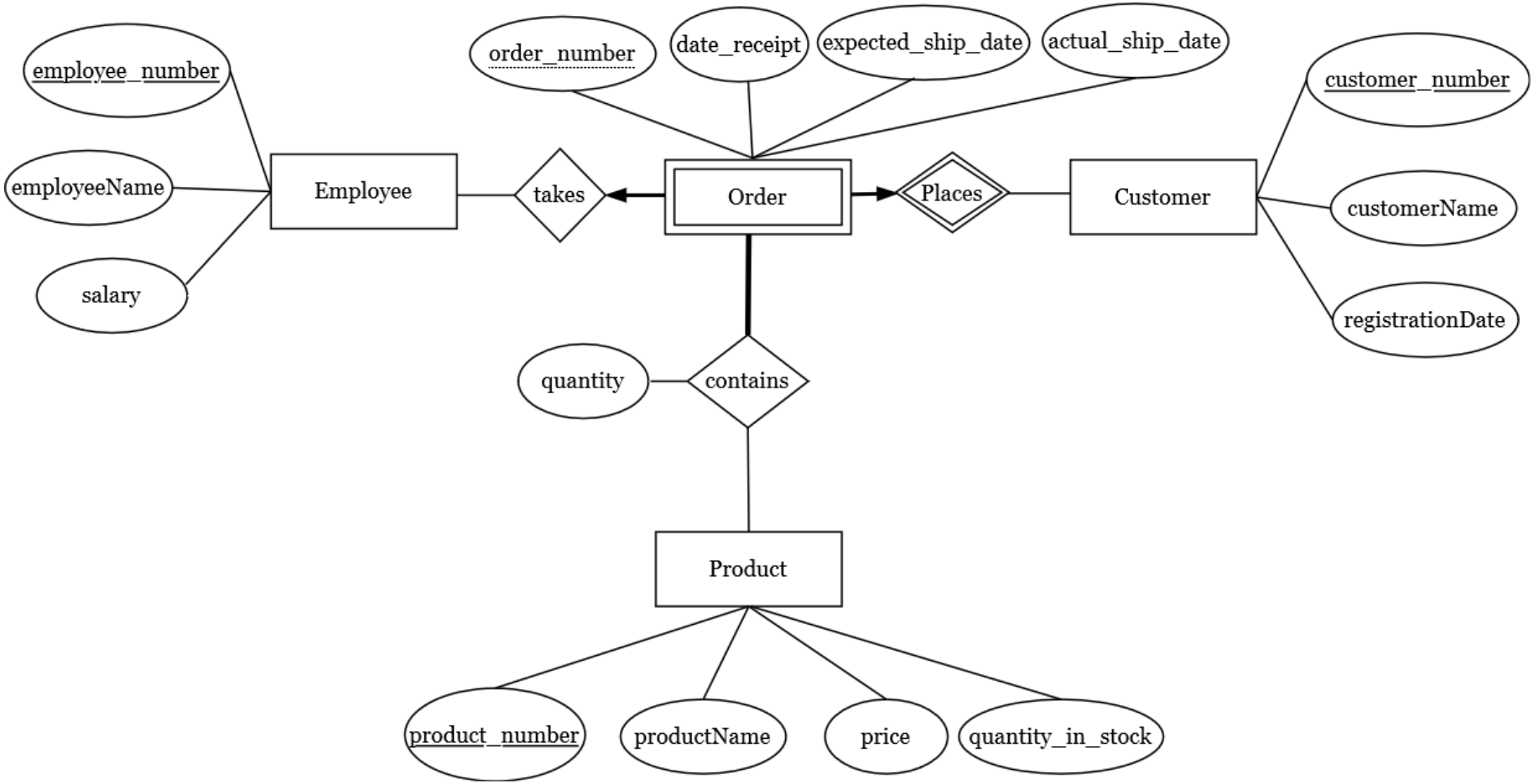


- Looking on each relationship side:
  - 1 library contains 0 to many books
  - 1 book can be found in 0 to many libraries



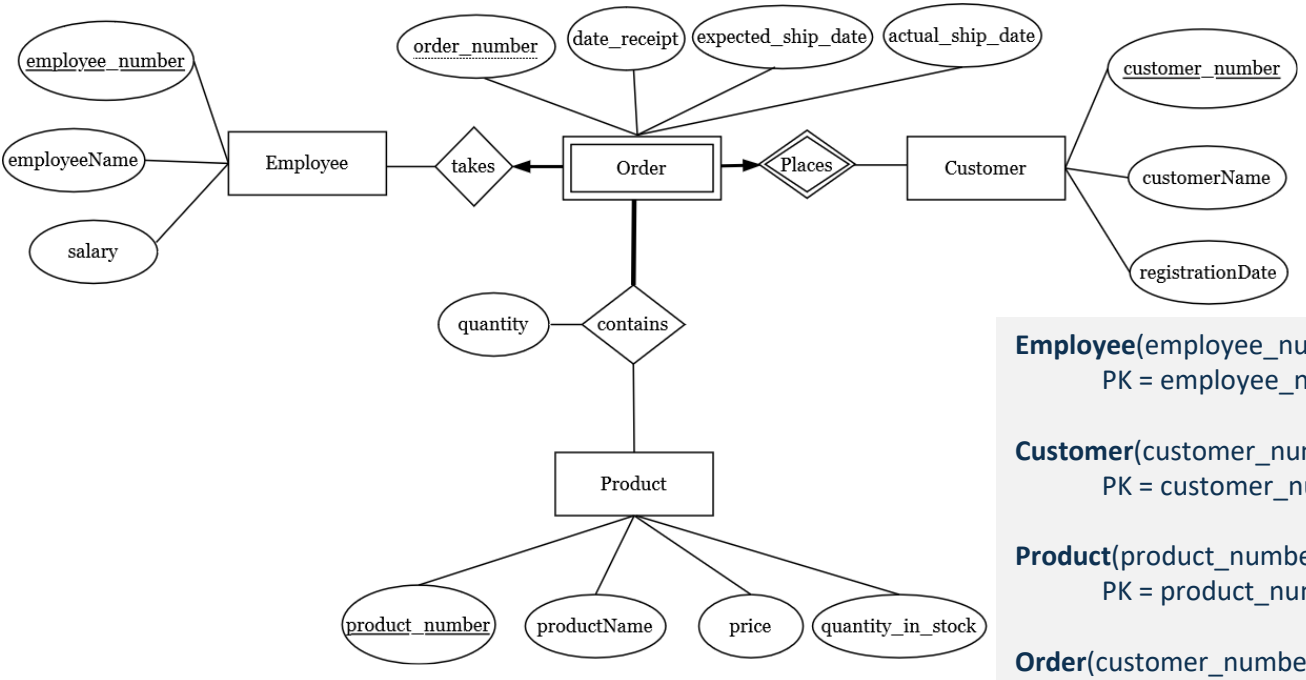
Creating Schema Diagram

Create the schema diagram equivalent to the following ER diagram



Creating Schema Diagram

Create the schema diagram equivalent to the following ER diagram



Employee(employee_number, employeeName, salary)
PK = employee_number

Customer(customer_number, customerName, registrationDate)
PK = customer_number

Product(product_number, productName, price, quantity_in_stock)
PK = product_number

Order(customer_number, order_number, date_receipt, expected_ship_date, actual_ship_date, employee_number)
PK = (customer_number, order_number)
FK = customer_number --> Customer
employee_number --> Employee

Contains(customer_number, order_number, product_number, quantity)
PK = (customer_number, order_number, product_number)
FK = customer_number --> Order
order_number --> Order
product_number --> Product

Student			
sid	**name**	**gender**	**country**
1001	Ian	M	AUS
1002	Ha Tschi	F	ROK
1003	Grant	M	AUS
1004	Simon	M	GBR
1005	Jesse	F	CHN
1006	Franzisca	F	GER

```
CREATE TABLE Student
(
    sid       INTEGER PRIMARY KEY,
    name      VARCHAR(20),
    gender    CHAR CHECK(gender IN ('F','M')),
    country   CHAR(3)
);
```

- Modify the table to set the name attribute as Not Null

  ALTER TABLE Student **ALTER COLUMN** name **SET NOT NULL**;

- Write a SQL statement to change the country of Simon from GBR to USA

  UPDATE Student **SET** country = 'USA' **WHERE** name = 'Simon'

- Write a SQL statement to delete all records from Student table

  DELETE FROM Student;

- Write a SQL statement to delete Student table

  DROP TABLE Student;

Relational Algebra

- **List the names of all students enrolled in 'DBMS'**
  - $\pi_{name}\ (\sigma_{title='DBMS'}\ ((Student \bowtie Enrolled) \bowtie UnitOfStudy))$

P[name] (S[title='DBMS'] ((Student J Enrolled) J UnitOfStudy))

Selection	σ	S	Union	∪	U
Projection	π	P	Intersection	∩	I
Cross-product	×	X	Difference	-	D
Join	⋈	J	Rename	ρ	R
Conditional Join	⋈Θ	CJ	AND	/\	and
			OR	\/	or

- **List the id of students who are not enrolled in any unit of study**

P[sid] (Student) - P[sid] (Enrolled)

Student(<u>sid</u>, name, gender, country)
Enrolled(<u>sid, uos_code,</u> semester)
UnitOfStudy(<u>uos_code</u>, title, points)

Relational Algebra

Suppliers(sid, sname, address)

Product(pid, pname, colour)

Catalog(sid, pid, price)

› Find the names of all suppliers who supply a product that is red or green.

$$\pi_{sname}((\pi_{sid}((\sigma_{colour='red' \vee colour='green'}(Product)) \bowtie Catalog)) \bowtie Suppliers)$$

› Find the id of all students who are enrolled in both ‘COMP5138’ and ‘COMP5318’.

```
SELECT sid FROM Enrolled WHERE uos_code=‘COMP5138’
INTERSECT
SELECT sid FROM Enrolled WHERE uos_code=‘COMP5318’
```

› Find the names of students who are enrolled in both ‘COMP5138’ and ‘COMP5318’

```
SELECT name
 FROM Student
WHERE sid IN ( SELECT sid FROM Enrolled WHERE uos_code=‘COMP5138’
               INTERSECT
               SELECT sid FROM Enrolled WHERE uos_code=‘COMP5318’
             )
```

Student(sid, name, gender, country)
Enrolled(sid, uos_code, semester)
UnitOfStudy(uos_code, title, points)

› Find the name of the students who are enrolled in the unit of study that has the highest credit points

Approach 1:

```
SELECT name
FROM Student
WHERE sid IN (SELECT sid
              FROM Enrolled
              WHERE uos_code IN (SELECT uos_code
                                 FROM UnitOfStudy
                                 WHERE points = (SELECT max(points) FROM UnitOfStudy )
                                )
             )
```

Approach 2:

```
SELECT name
FROM Student NATURAL JOIN Enrolled NATURAL JOIN UnitOfStudy
WHERE points = ( SELECT max(points) FROM UnitOfStudy)
```

Student(sid, name, gender, country)
Enrolled(sid, uos_code, semester)
UnitOfStudy(uos_code, title, points)

› Find the number of students in each country, except Australia ('AUS'). Only include countries that have equal or more students than Australia. Show the output in sorted order of country name.

```
SELECT country, COUNT(sid)
FROM Student
WHERE country <> 'AUS'
GROUP BY Country
HAVING COUNT(sid) >= (SELECT COUNT(sid) FROM Student WHERE country = 'AUS')
ORDER BY country
```

Student(sid, name, gender, country)
Enrolled(sid, uos_code, semester)
UnitOfStudy(uos_code, title, points)

… and that's it.

Unit of Study Survey

Your Unit of Study Survey (USS) feedback is **confidential**.

It's a way to share what you enjoyed and found most useful in your learning, and to provide constructive feedback. It's also a way to 'pay it forward' for the students coming behind you, so that their **learning experience** in this class is as good, or even better, than your own.

When you complete your USS survey, please:

Be specific

Which class tasks, assessments or other activities helped you to learn? Why were they helpful? Which one(s) didn't help you to learn? Why didn't they work for you?

Be constructive.

What practical changes can you suggest to class tasks, assessments or other activities, to help the next class learn better?

Be relevant.

Imagine you are the teacher. What sort of feedback would you find most useful to help make your teaching more effective?

https://student-surveys.sydney.edu.au

Best of luck!

THE UNIVERSITY OF SYDNEY