



MONASH  
University

MONASH  
INFORMATION  
TECHNOLOGY

# Database Design 1: Conceptual Modelling

FIT2094-FIT3171

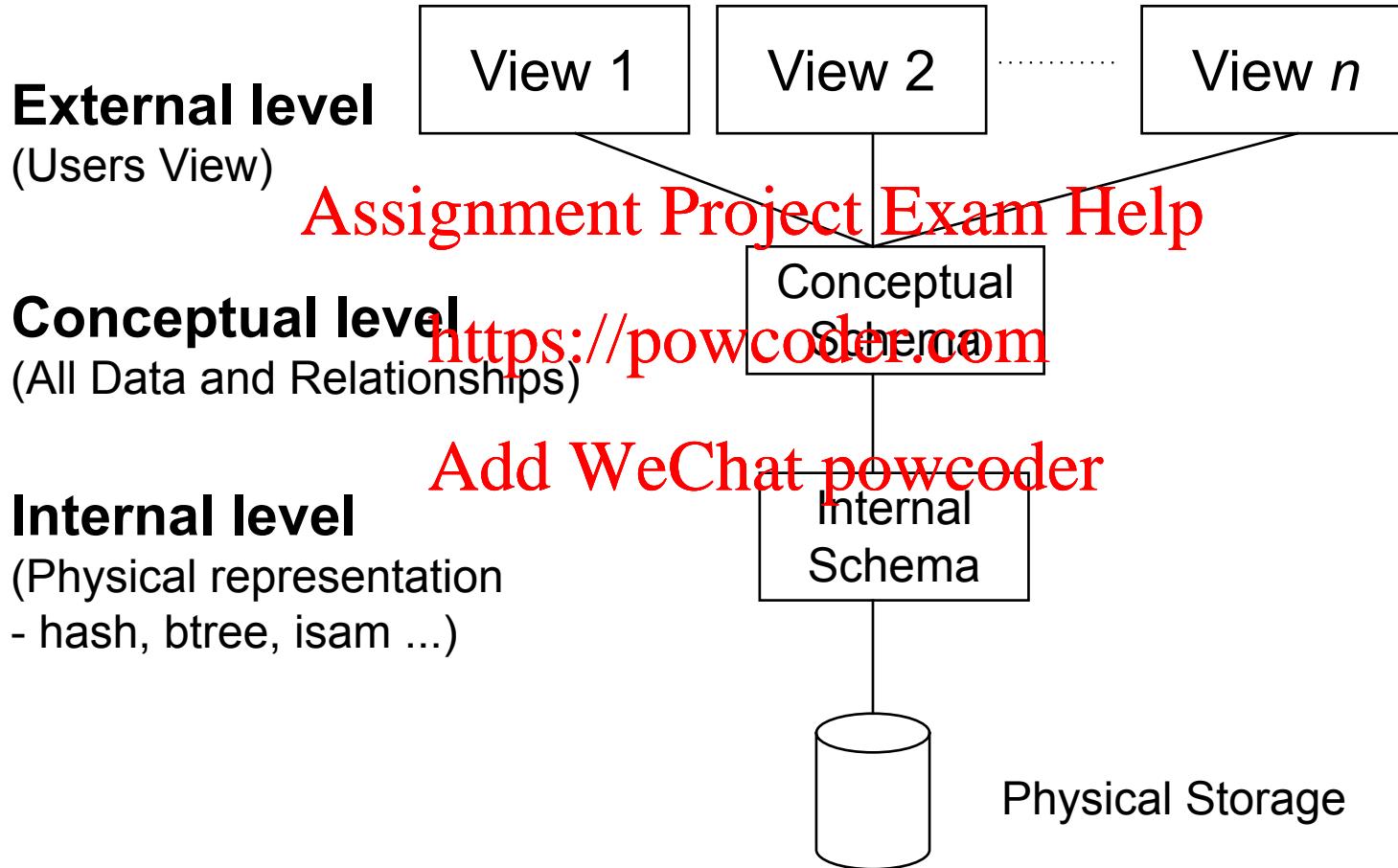
Assignment Project Exam Help

<https://powcoder.com>

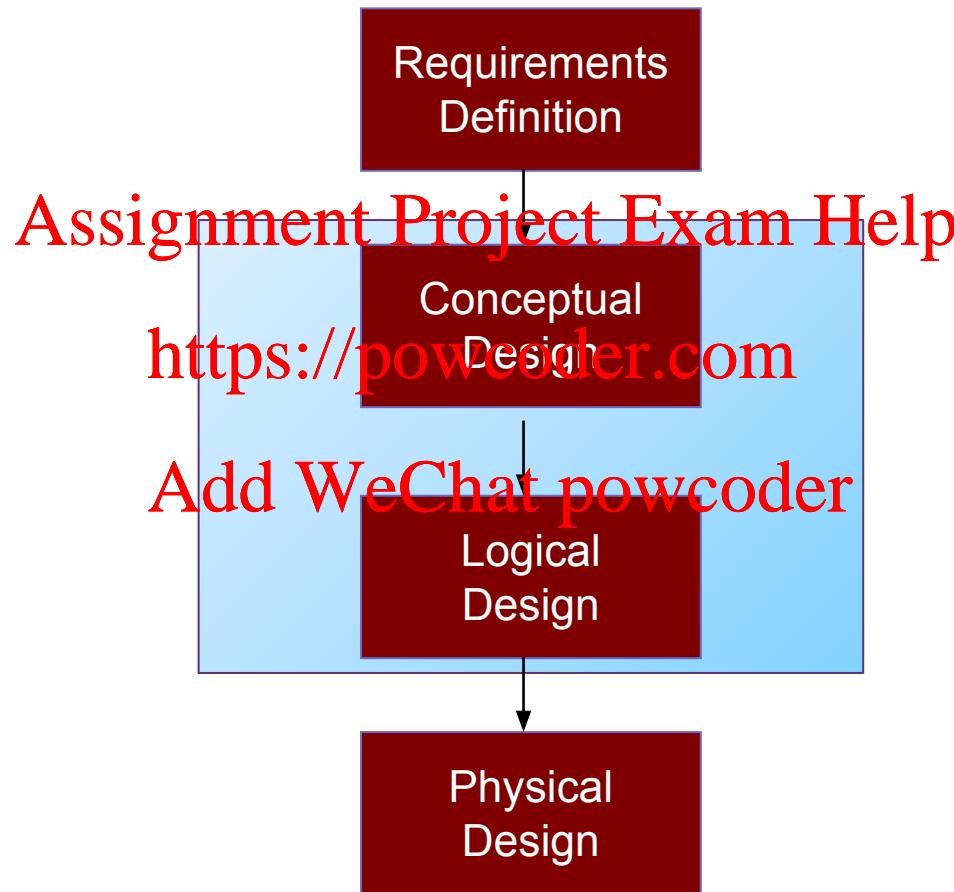
Add WeChat powcoder



# ANSI/SPARC architecture - proposed 1975



# The Database Design Life Cycle



# Requirements Definition

- Identify and analyse user views.
- A 'user view' may be a report to be produced or a particular type of transaction that should be supported.  
<https://powcoder.com>
- Corresponds to the external level of the ANSI/SPARC architecture.  
[Add WeChat: powcoder](https://powcoder.com)
- Output is a statement of specifications which describes the user views' particular requirements and constraints.

# Different views of the underlying data

**Web Enrolment System**

- Enrolment / Re-Enrolment
- Fees / Scholarships
- Student Services
- Course Progression
- Enrolment Access Dates
- WES Guides
- Monash Links**
- my.monash
- Allocate+ (Class Allocation)
- Class timetable (prev. MUTTS)
- Moodle

Student						
To Add Units Click here						
Unit code	Action	Unit name	Campus	Semester	Type	Credits
ACF1200	Change Remove	Accounting for managers <b>PENDING - ENROLLED</b>	CAUL	Semester 1 (2018)	ON-CAMPUS	6
BFF1001	Change Remove	Foundations of finance <b>PENDING - ENROLLED</b>	CAUL	Semester 1 (2018)	ON-CAMPUS	6
BTF1010	Change Remove	Business law <b>PENDING - ENROLLED</b>	CAUL	Semester 1 (2018)	ON-CAMPUS	6
MKF1120	Change Remove	Marketing theory and practice <b>PENDING - ENROLLED</b>	CAUL	Semester 1 (2018)	ON-CAMPUS	6
ECF1100	Change Remove	Microeconomics <b>PENDING - ENROLLED</b>	CAUL	Semester 2 (2018)	ON-CAMPUS	6
ETF1100	Change Remove	Business statistics <b>PENDING - ENROLLED</b>	CAUL	Semester 2 (2018)	ON-CAMPUS	6
MGF1010	Change Remove	Introduction to management <b>PENDING - ENROLLED</b>	CAUL	Semester 2 (2018)	ON-CAMPUS	6

Allocate<sup>+</sup>

Activity Groups:

► **Applied\_PASS**  
(READ ONLY)  
Enrolments: 826  
Preferences: 0  
Allocations: 0  
Seats Provided: 180  
Warning: Not enough seats provided

► **Tutorial**  
(ALLOCATION ADJUSTMENT)  
Enrolments: 826  
Preferences: 260  
Allocations: 826  
Seats Provided: 840

► **Workshop**  
(ALLOCATION ADJUSTMENT)  
Enrolments: 826  
Preferences: 259  
Allocations: 826  
Seats Provided: 840

**Subject Administrator**  
**FIT9132\_CL\_S1\_ON-CAMPUS, INTRO TO DATABASES**  
**Tutorial**

Allocate    Add Activity    Show Message    Show Allocated    Waitlist    Show Unallocated

Functions		Activity Code	Campus	Day	Start Time	Location	Staff	Duration	Planned Size	Buffer
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		01	CL	Tue	12:00	CL_20Exh/G15	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		02	CL	Tue	14:00	CL_20Exh/G15	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		03	CL	Tue	16:00	CL_20Exh/G05	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		04	CL	Mon	18:00	CL_20Exh/104	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		05	CL	Wed	14:00	CL_20Exh/G05	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		06	CL	Thu	08:00	CL_20Exh/G15	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		07	CL	Thu	10:00	CL_20Exh/G15	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		08	CL	Mon	08:00	CL_20Exh/104	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		09	CL	Tue	08:00	CL_20Exh/107	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		10	CL	Tue	18:00	CL_20Exh/G05	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		11	CL	Mon	16:00	CL_20Exh/G06	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		12	CL	Mon	10:00	CL_20Exh/104	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		13	CL	Mon	12:00	CL_20Exh/107	-	120	60	0
<a href="#">Delete</a>   <a href="#">Edit</a>   <a href="#">List</a>   <a href="#">Constraint</a>   <a href="#">Context</a>   <a href="#">Email</a>		14	CL	Mon	14:00	CL_20Exh/G05	-	120	60	0

## Unit guides



### FIT9132: Introduction to databases

Semester 1 (S1-01) 2020



We acknowledge and pay respects to the Traditional Owners and Elders - past, present and emerging - of the lands and waters on which Monash University operates.

The information contained in this unit guide is correct at time of publication. The University has the right to change any of the elements contained in this document at any time.

Last updated: 23 Feb 2020

Status: Approved

#### Table of contents

#### Unit handbook information

##### Synopsis

This unit will introduce the concept of data management in an organisation through relational database technology. Theoretical foundation of relational model, analysis and design, of relational database using SQL will be covered.

#### Mode(s) and mode(s) of delivery

on-campus

#### Ad requirements

If expected workload equals 12 hours per week comprising:  
hours for on-campus students:

Staff &  
Student

Admin

# ER Modeling

- ER (Entity-Relationship) model developed by Peter Chen in 1976 to aid database design.
- Used for [Assignment Project Exam Help](#)
- ER diagrams give a visual indication of the design.  
<https://powcoder.com>
- Basic components:
  - Entity
  - Attribute
  - Relationship

Add WeChat powcoder



# Conceptual Design

- Develop the enterprise data model.
- Corresponds to the conceptual level of the ANSI/SPARC architecture.  
**Assignment Project Exam Help**  
**https://powcoder.com**
- *Independent of all physical implementation considerations (the type of database to be used).*
- Various design methodologies may be employed such as UML, ER (Entity-Relationship).
- ER consists of ENTITIES and RELATIONSHIPS between entities
  - An ENTITY will have attributes (things we wish to record), one or more of which will identify an entity instance (called the KEY)

# Conceptual Level (ER Model)

## ENTITY

*Collection of "Customer(s)"*

CUSTOMER	
Key	custno
	custname
	custaddress
	custphone

## RELATIONSHIP

*Relates entities*

ORDER	
Key	orderno
	orderdate

## KEY ATTRIBUTE(S)

*Instance identifier*

## NON KEY ATTRIBUTE

*Other non-key attributes*

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment 1A

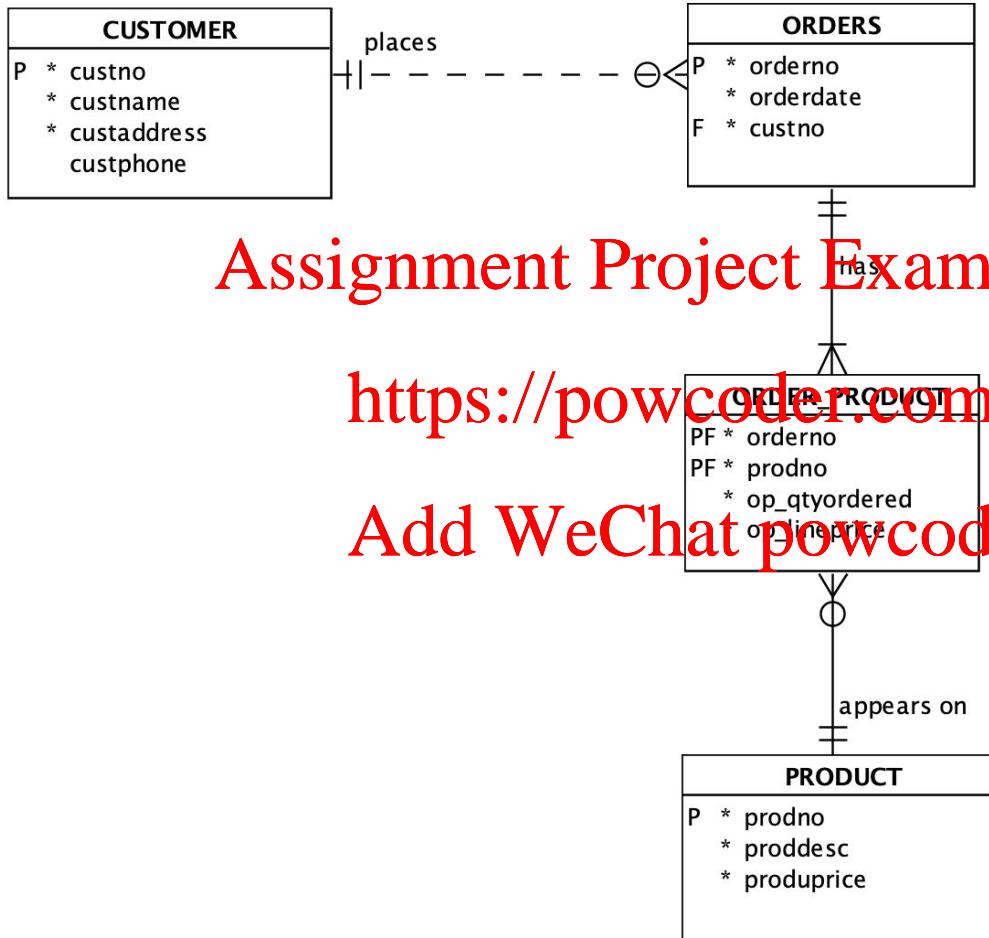
ORDER-PRODUCT	
Key	orderno
Key	prodno
	op_qtyordered
	op_lineprice

PRODUCT	
Key	prodno
	proddesc
	prodprice

# Logical Design

- Develop a data model which targets a particular database model (e.g. relational, hierarchical, network, object-oriented, noSQL).  
**Assignment Project Exam Help**  
**https://powcoder.com**
- Independent of any implementation details which are specific to any particular vendor's DBMS package.
- Normalisation  
**Add WeChat powcoder** (see week 4) is used to test the correctness of a relational logical model.

# Logical Level (Logical Model - Relational)



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Assignment 1B

# Physical Design

- Develop a strategy for the physical implementation of the logical data model.
- Choose appropriate storage structures, indexes, file organisations and access methods which will most efficiently support the user requirements (not part of unit).
- Physical design phase is dependent on the particular DBMS environment in use.
- ANSI/SPARC internal level.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

# Physical Level – Starting point

Oracle Database 12c Relational\_1 Generate

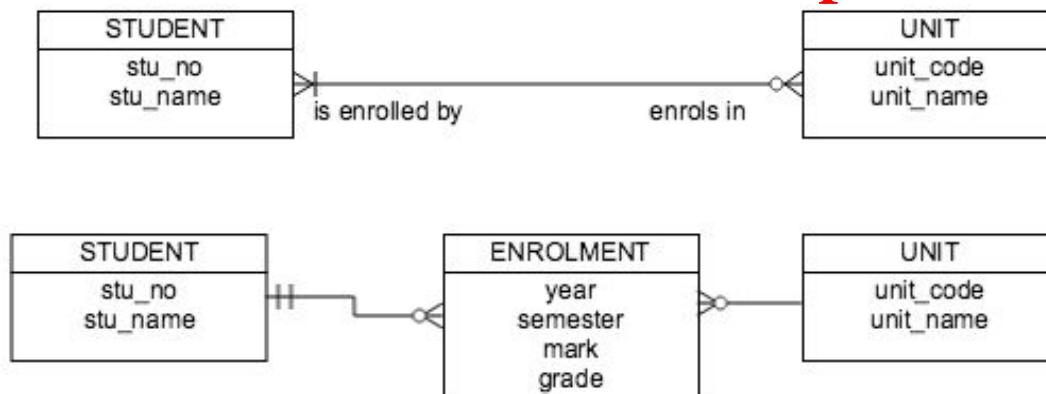
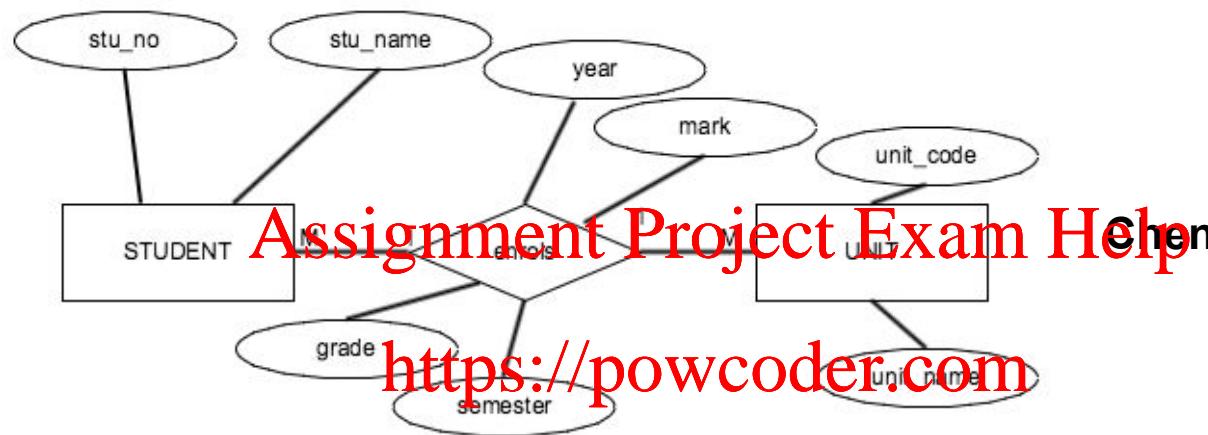
```
8  CREATE TABLE customer (
9      custno      NUMBER(7) NOT NULL,
10     custname     VARCHAR2(50) NOT NULL,
11     custaddress  VARCHAR2(50) NOT NULL,
12     custphone    CHAR(10)
13 );
14
15 COMMENT ON COLUMN customer.custno IS
16   'Customer number';
17
18 COMMENT ON COLUMN customer.custname IS
19   'Customer name';
20
21 COMMENT ON COLUMN customer.custaddress IS
22   'Customer address';
23
24 COMMENT ON COLUMN customer.custphone IS
25   'Customer phone number';
26
27 ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY ( custno );
28
29 CREATE TABLE order_product (
30     orderno      NUMBER(7) NOT NULL,
31     prodno       NUMBER(7) NOT NULL,
32     op_qtyordered NUMBER(3) NOT NULL,
33     op_lineprice  NUMBER(8, 2) NOT NULL
34 );
35
```

Assignment Project Exam Help

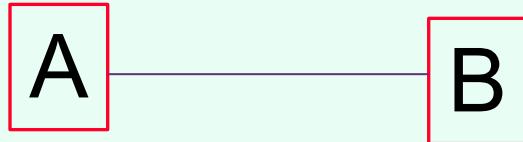
<https://powcoder.com>

Add WeChat powcoder

# ERD - Notation



Information  
Engineering/James  
Martin/Crows foot  
*\* This is what we will  
be using*



*Please note this diagram is incomplete*

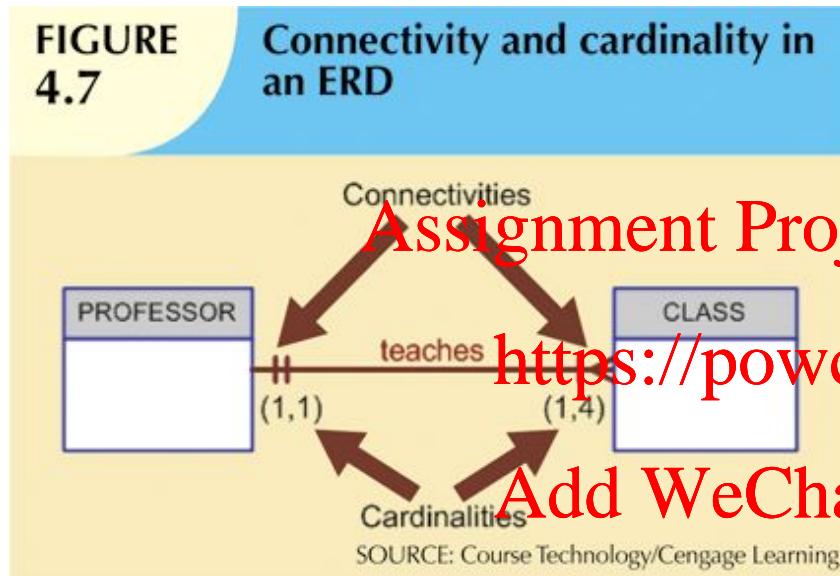
**Q1. How many relationships are there in the above diagram?**

**Assignment Project Exam Help**  
What is the degree of the relationship/s (the number of entities participating in the relationship/s)?  
Add WeChat powcoder

- A. 1, unary
- B. 2, binary
- C. 1, binary
- D. 3, ternary

**FIGURE  
4.7**

Connectivity and cardinality in  
an ERD



Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

*In general for Crows Foot notation specific cardinalities are not shown as above eg. (1,4), instead participation is depicted via min and max participation using the standard symbols (Inside symbol = min, outside symbol = max)*

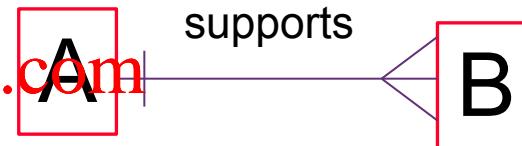
**CONNECTIVITY**

**one to one**



supports

**one to many**



supports

**many to many**



supports

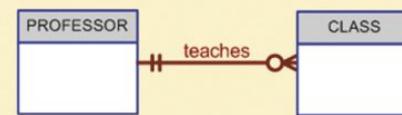
FIGURE  
4.15

Three types of relationship degree

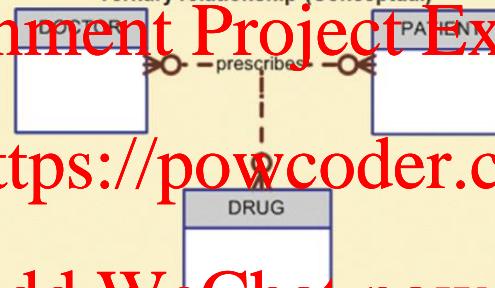
Unary relationship



Binary relationship



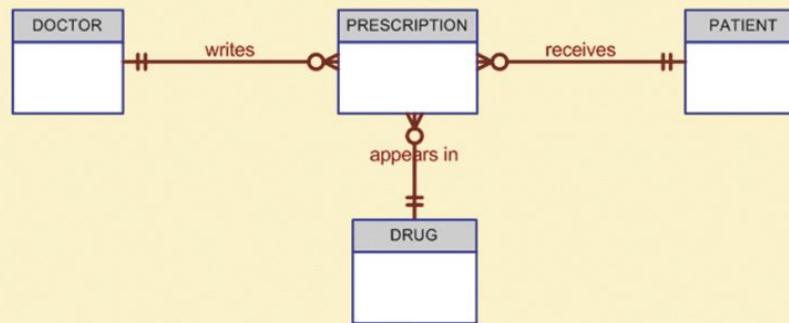
Ternary relationship (Conceptual)



Assignment Project Exam Help  
<https://powcoder.com>

Add WeChat powcoder

Ternary relationship (Logical)



SOURCE: Course Technology/Cengage Learning

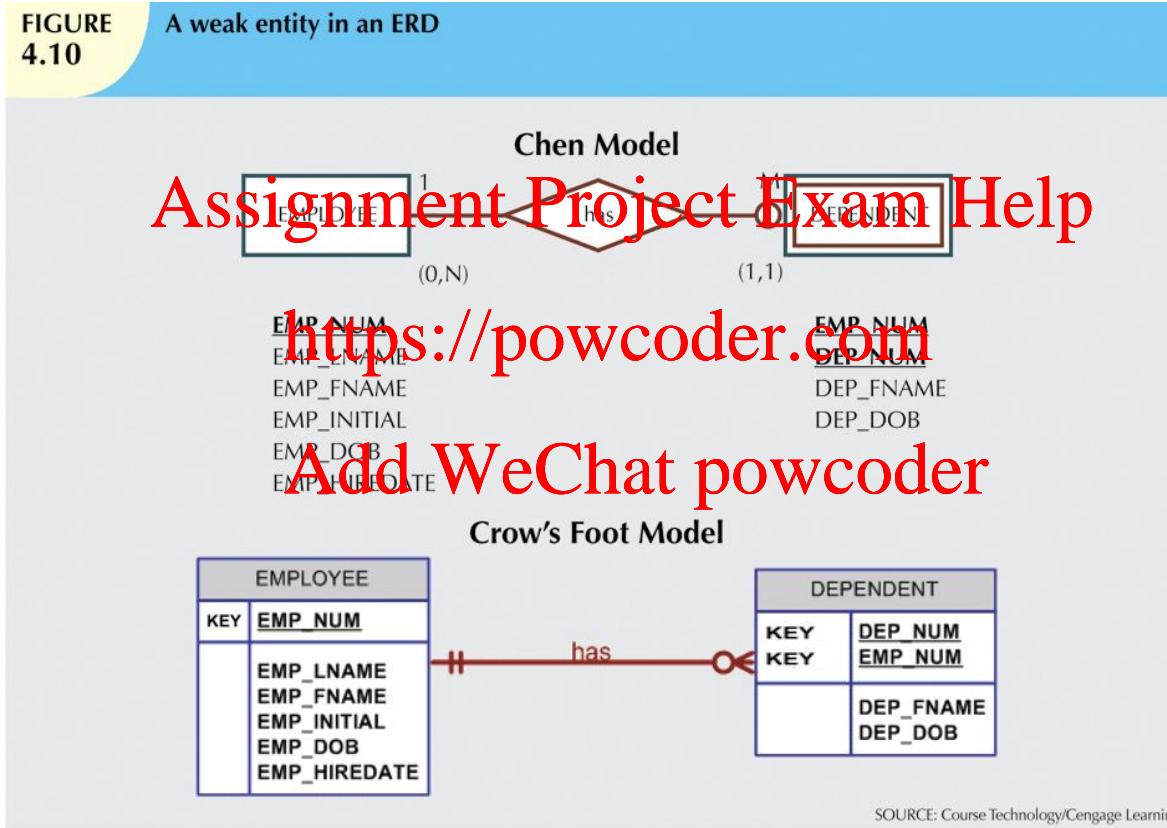
# Weak vs Strong Entity

- Strong entity
  - Has a key which may be defined without reference to other entities.
  - For example EMPLOYEE entity.
- Weak entity **Assignment Project Exam Help**
  - Has a key which requires the existence of one or more other entities.
  - For example FAMILY entity – need to include the key of employee to create a suitable key for family
- Database designer often determines whether an entity can be described as weak based on business rules
  - customer pays monthly account
    - Key: cust\_no, date\_paid, or
    - Key: payment\_no (surrogate? – not at conceptual level)

# Weak vs Strong Entity

FIGURE  
4.10

A weak entity in an ERD



*Note the Crow's Foot model shown here has been modified from the text version*

# Identifying vs Non-Identifying Relationship

- **Identifying**

- Identifier of A is part of identifier of B.

- **Non-identifying**

- Identifier of A is NOT part of identifier of B.

Assignment Project Exam Help



- Shown with solid line
- ENROLMENT - STUDENT  
Enrolment key includes student id, which is an identifier of student.

- Shown with broken line
- Department no (identifier of department) is not part of Employee's identifier.

## Student Activities

- Working in teams, using the Monash Software Case Study, identify the entities present and their key attribute/s  
**Assignment Project Exam Help**
  - draw a box for each entity, name it and add the key attribute/s
  - For example: <https://powcoder.com>



# Entities in the Monash Software Case Study

TRAINING	
Key	training_code

Assignment Project Exam Help

<https://powcoder.com>

TEAM	
Key	team_no

Add WeChat powcoder

EMPLOYEE	
Key	emp_no

FAMILY_MEMBER	
Key	

# Student Activities

- Working in teams, using the Monash Software Case Study, identify the relationship(s) and participation which exist between TEAM and EMPLOYEE

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

TEAM	
Key	team_no

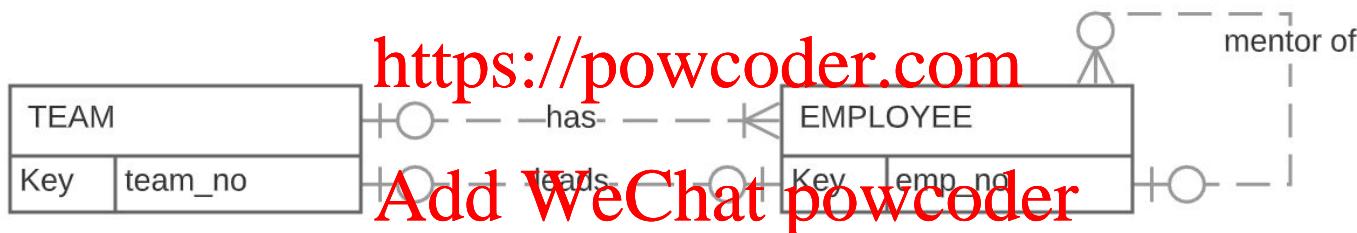
EMPLOYEE	
Key	emp_no

FAMILY_MEMBER	
Key	

# TEAM - EMPLOYEE relationships

TRAINING	
Key	training_code

Assignment Project Exam Help



FAMILY_MEMBER	
Key	

# Types of Attributes

- Simple
  - Cannot be subdivided
  - Age, sex, marital status
- Composite
  - Can be subdivided into additional attributes
  - Address into street, city, zip
- Single-valued
  - Can have only a single value
  - Person has one social security number
- Multi-valued
  - Can have many values
  - Person may have several college degrees
- Derived
  - Can be derived with algorithm
  - Age can be derived from date of birth
- Attribute classification is driven by Client requirements
  - Phone Number?

**Q2. The employee details that will be recorded are:**

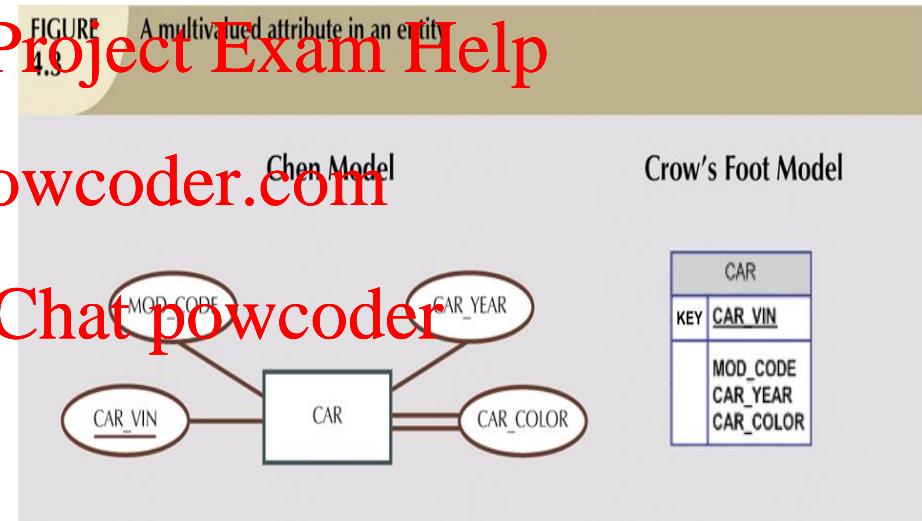
**Employee number, Full name, Address, Date of birth, Tax file number and Skill(s). Examples of skills are Java, Python, UNIX, Relational db, MongoDB, etc**

**Choose a TRU Assignment Project Exam Help**

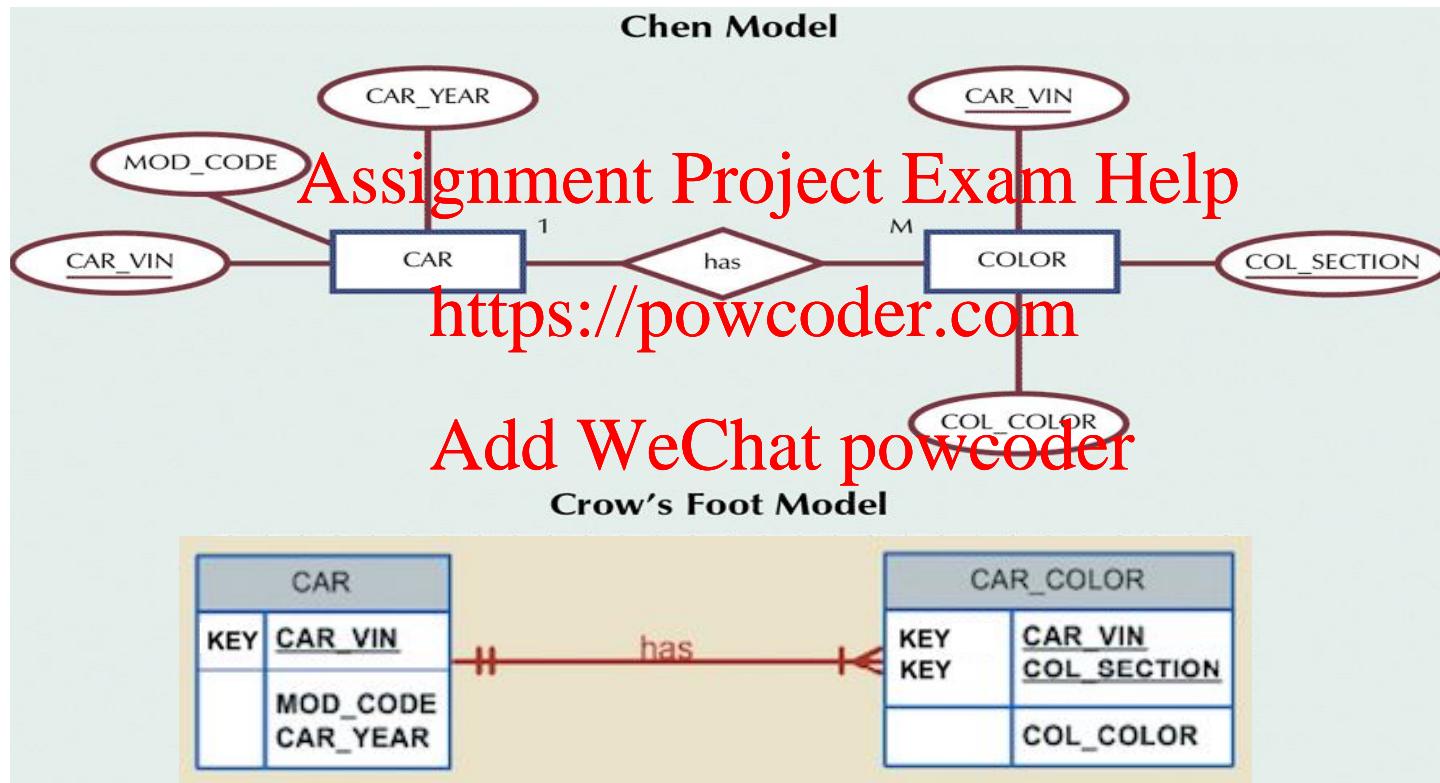
- A. ADDRESS is a multi-valued attribute and SKILL is a multi-valued attribute.
- B. ADDRESS is a composite attribute and SKILL is a composite attribute.
- C. ADDRESS is a composite attribute and SKILL is a multi-valued attribute.
- D. ADDRESS is a multi-valued attribute and SKILL is a composite attribute.

# Multivalued Attribute

- An attribute that has a list of values.
- For example:
  - Car colour may consist of body colour, trim colour, bumper colour.
- Crow's foot notation does not support multivalued attributes. Values are listed as a separate attribute.



# Resolving Multivalued Attributes



# Student Activities

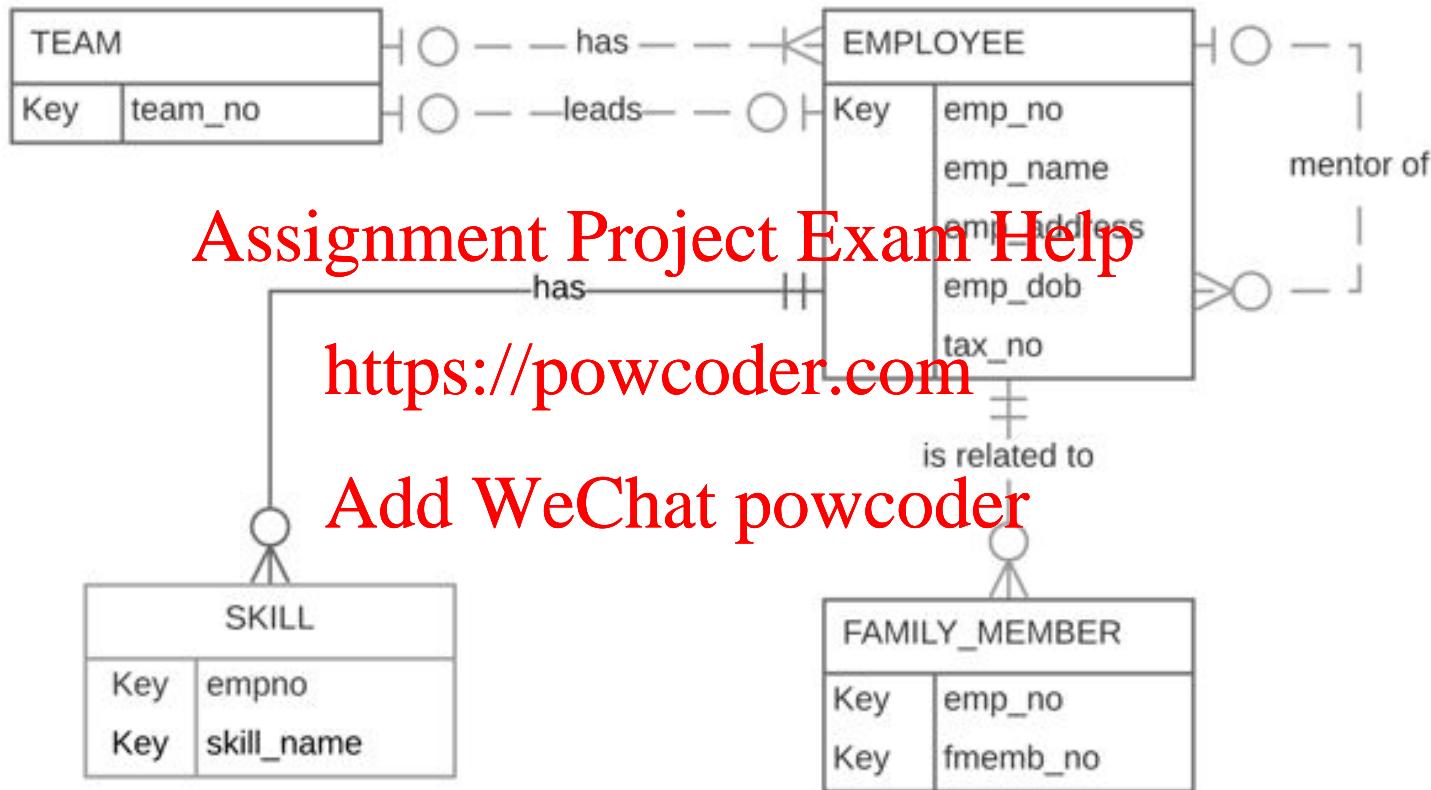
- Working in teams, using the Monash Software Case Study, add attributes to your EMPLOYEE entity

Assignment Project Exam Help



<https://powcoder.com>





**Q3. The company provides several in-house training programs. The HR team needs to keep track of the details about who has done what. An employee can do several training programs. At the completion of a training, a certificate will be provided to the employee containing the training name.**

- A. The relationship between EMPLOYEE and TRAINING is ternary.
- B. The relationship between EMPLOYEE and TRAINING is 1:M.
- C. The relationship between EMPLOYEE and TRAINING is M:1.
- D. The relationship between EMPLOYEE and TRAINING is M:N.

## Incomplete model

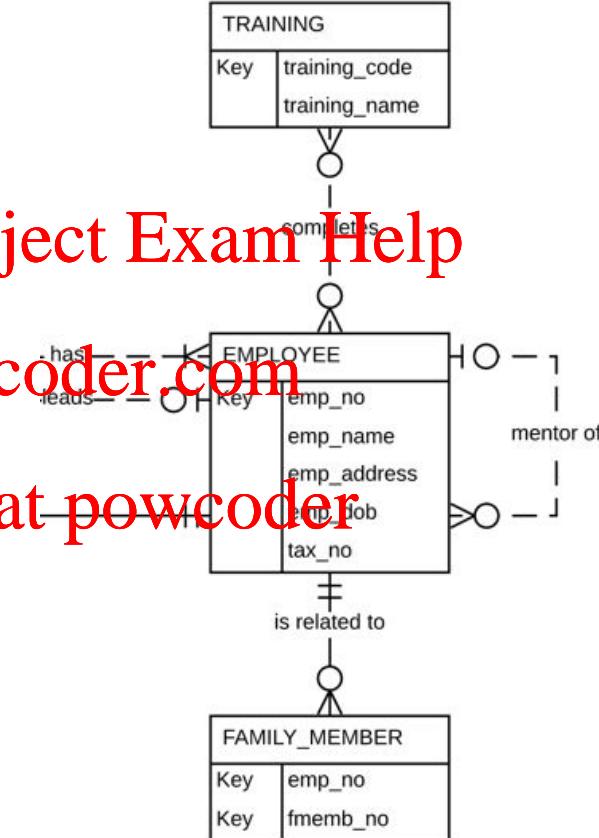
*... "the company provides several in-house training programs. The HR team needs to keep track of the details about who has done what. An employee can do several training programs.*

Assignment Project Exam Help

<https://powcoder.com>

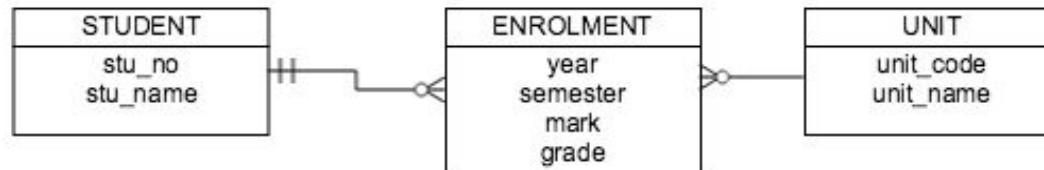
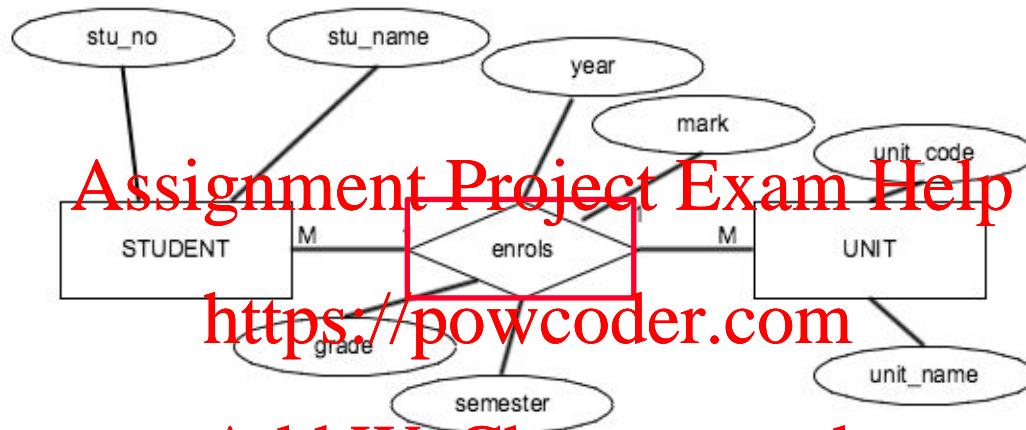
*At the completion of a training, a certificate will be provided to the employee containing the training name and the completion date."...*

Add WeChat powcoder



*Incomplete model*

# Associative (or Composite) Entity



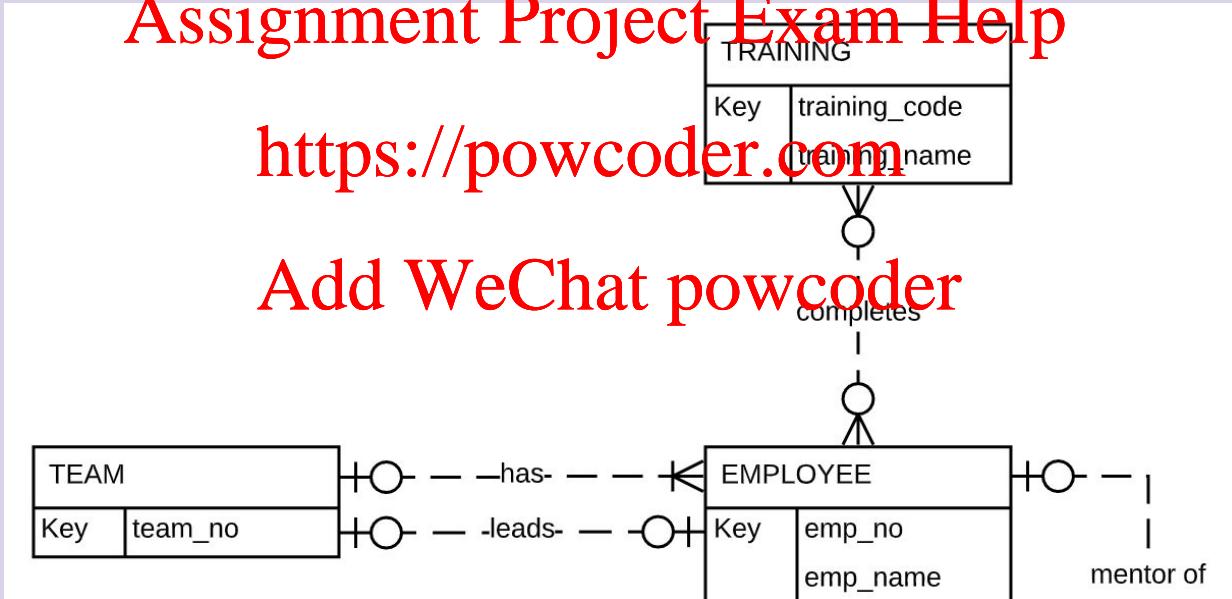
# Student Activities

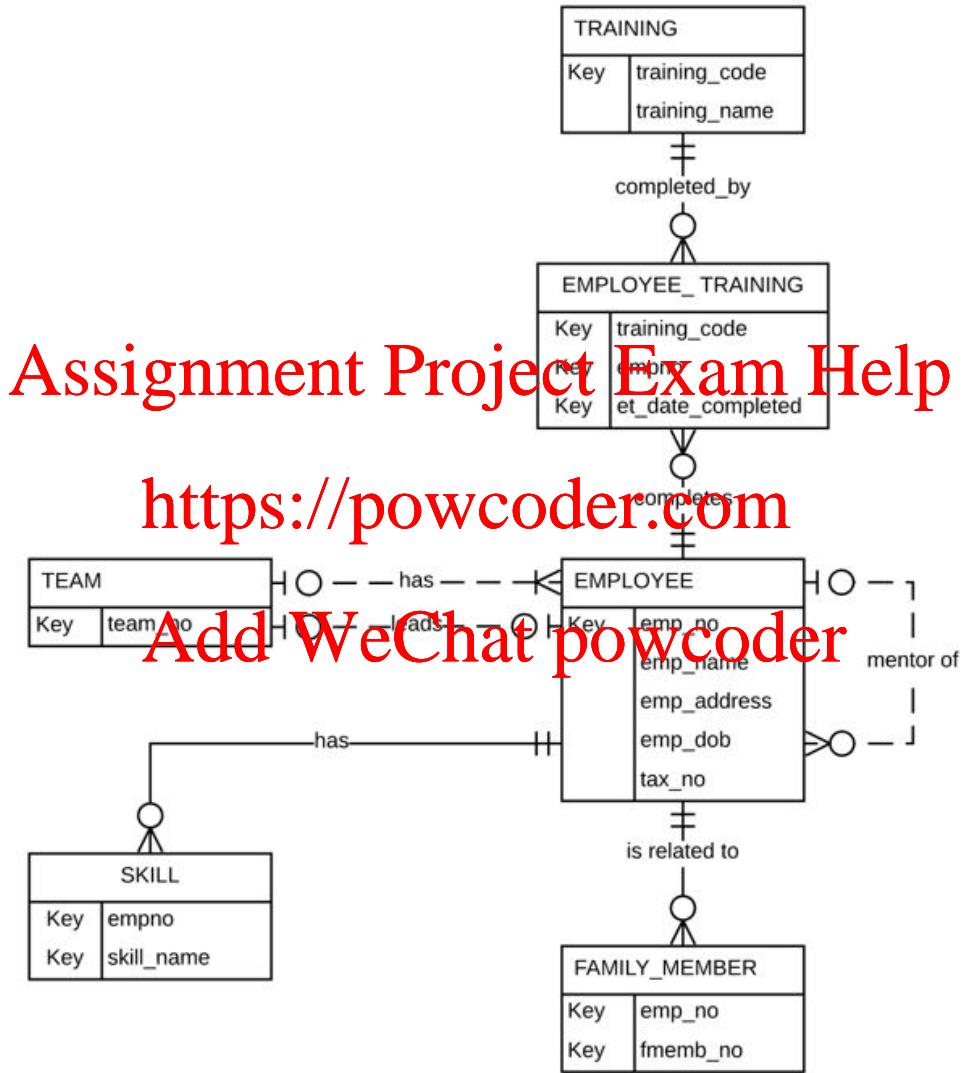
- Working in teams, using the Monash Software Case Study, add attribute to TRAINING EMPLOYEE

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder





# Unified Modeling Language (UML)

- No assessment on UML for FIT2094
- FIT3171 will be examined on UML:
  - submit a UML diagram along with crow's foot ER Diagram in Assignment 1a **Assignment Project Exam Help**
  - some questions in exam **<https://powcoder.com>**

Add WeChat powcoder

# Unified Modeling Language (UML)

- The way that data is organised in a database is very different to the way it is organised in an OO program
  - eg. inheritance
- Use a **subset** of UML notation for database modelling
- Several vendors support Database Modelling via UML,  
<https://powcoder.com>
  - Star UML
  - Altova UModel
- Variety of standards adopted, not widely used in practice

# UML Notation for the unit

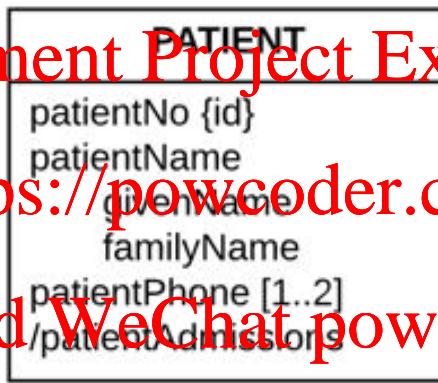
Standard UML Diagram is used as the basic structure:

UML Class Name
attribute names
UML Methods (not needed for database modelling)

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

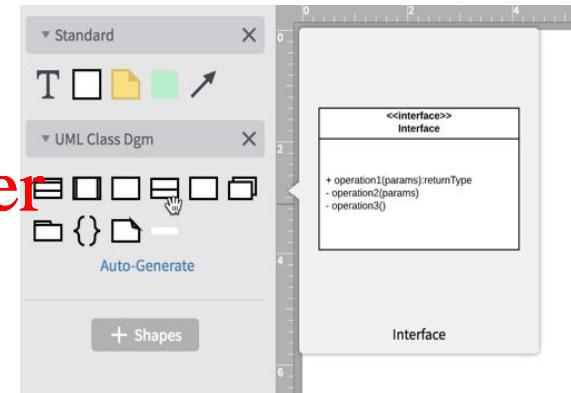


{id} - indicates KEY

Indentation for composite attribute

[n..m] - multivalued attribute

/ - calculated attribute



use Interface shape as no methods being added

The employee details that will be recorded are:

- Employee number
- Full name
- Address
- Date of birth
- Tax file number
- Skill(s). For example, Java, Python, UNIX, Relational DB, Mongo DB, etc

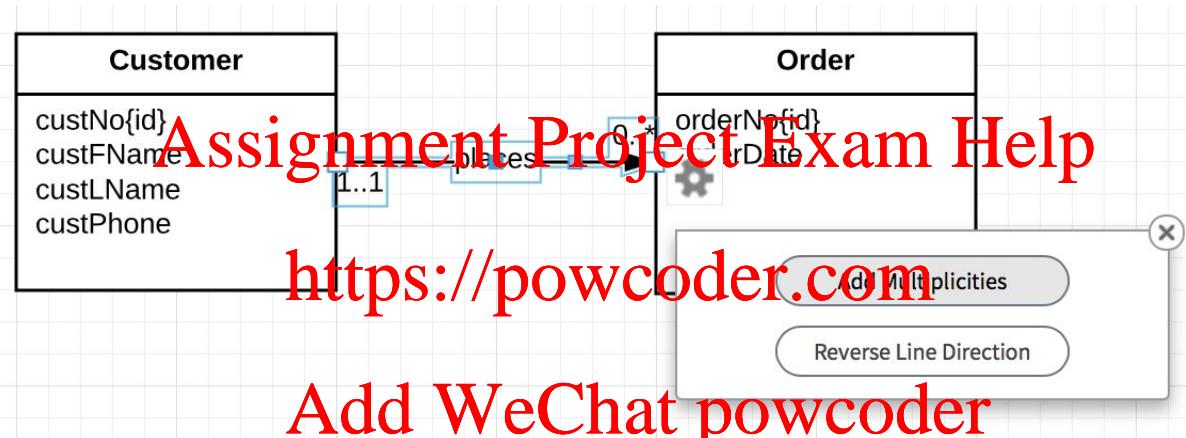
Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EMPLOYEE
empNo {id}
empName
firstName
lastName
empAddress
street
town
postocde
empDOB
empTaxNo
empSkill [0..5]

# UML - Relationship



Relationship lines - directed line, arrowhead at M end,  
add Multiplicities (minimal and maximal on each side)

