Tutorial 1 & 2

1. Explain TWO (2) application areas of database system AARON WONG ZYI SYEN

Airlines and Railways-It is used for reservation and displaying schedule information.

Bankings- It is used for customer enquiry, accounts, loans and other transactions.

1. Recommend **TWO (2)** factors when choosing the types of database. CECILIA KONG XIN RU

**Number of users**

Types of the database are chosen depending on the number of users that needed to support or operate at the same time.

**Amount of data/Number of transactions**

The types of database are chosen by determining the number of transactions needed or occured in the database so that the database operation can be committed .

**Hardware components and specifications**

The style and speed of computer hardware available.

**Method of access database**

Users should choose the types of database wisely in order to create and access their database which is suitable to their method for access databases such as using an office network , internet or hand held or wireless computer to prevent database failure or losing of data because of unsuccessful commit.

1. File-based systems were first developed to store, manipulate and retrieve data but as business applications become more complex, it was evident to have a number of limitations. Describe any **TWO (2)** disadvantages of using a file-based system. CHENG

cai jie

* Data Inconsistency: Different departments have different versions of similar data, for example one department updates the data but another department did not update that data.

- Data Redundancy (Data duplication): same information being kept in several different places

-Program-data dependence: program-data dependence exists when changes in data characteristic require changes in the program that access those data

- Excessive Program Maintenance: the structure of the data file is coupled with the individual application program. Therefore, any modification to a data file such as size of data field, its type etc. requires the modification of the application program also. this process of modifying program is referred to as program maintenance.(time consuming)

-Limited Data Sharing: files are not shared with other applications or among users. each application program uses its own private data files. The computer file-based processing system does not provide the facility to share data of a data file among multiple users on the network.

1. How a Database system can overcome the problems of File-Based system? CHI PUI MUN

-Data duplication: a database does not have uncontrolled redundancy.There is only one database managed by the DBMS that will ensure the database is updated and always consistent.

-Program-data dependence: When using a database and DBMS, change to a file/table/database structure would require none or minimal efforts in reprogramming and testing. Program and data are independent, especially in a 3-tier(or more) architecture.

-Excessive Program Maintenance: Programs can be maintained easily, changes to program logic does not affect database structure and vice-versa.

-Limited Data Sharing: the DBMS will allow data sharing among all authorised data.Different outlets can have different applications running but still able to connect to the database and share data.

1. Describe FIVE (5) main components needed in database system environment. CHUAH SHEE YEAP

Software

* Software is the set of program used to control and manage overall database

Hardware

* Consists of a set of physical electronic devices such as computers, input/output devices, storage devices, and etc.

Data

* The database contains both the actual or operational data (end-user data) and the metadata.

People

* the persons enter the data to and access from the database; to enter new data , updating existing data or retrieve required data from databases.

Procedures

* the instructions and rules that assist on how to use the DBMS, designing and running the database, using documented procedures.

1. A database management system (DBMS) performs several important functions to ensure the integrity and consistency of a data in the database. Briefly explain any FOUR (4) functions of a DBMS. GOH KHAI JUN

* Data dictionary management

Defines data elements and their relationships.

* Data storage management

Stores data and related data entry forms, report definitions, etc.

* Data transformation and presentation

Translates logical requests into commands to physically locate and retrieve the

requested data.

* Security management

To enforce the user security and data privacy within the database.

* Multi-user access control

Creates structures that allow multiple users to access the data.

* Backup and recovery management

Provides backup and data recovery procedures.

* Data integrity management

Promotes and enforces integrity rules to eliminate data integrity problems.

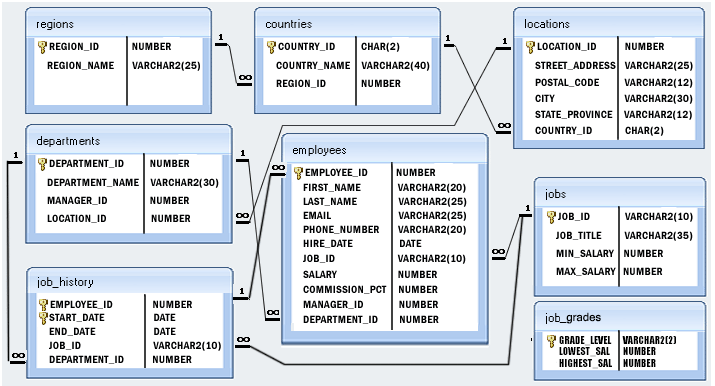
* Database access languages and application programming interfaces

Provides data access through a query language.

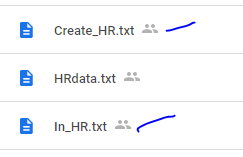
* Database communication interfaces

Allows database to accept end-user requests within a computer network environment.

**Tutorial 3 SQL**

The **HR** Entity Relationship Diagram

2 files needed: Create Table, InsertData.



Write **Structured Query Language (SQL) queries** to do the following:

Simple queries

1. Show the manager id, department id and department name for each department. HAR CHUN WAI

SELECT DEPARTMENT\_ID, DEPARTMENT\_NAME, MANAGER\_ID

FROM departments;

1. Show all employees with the first name ‘Alexander’. HO JING XIAN

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY

FROM employees

WHERE FIRST\_NAME = ‘Alexander’;

1. Show employee numbers, last names, salaries, and managers’ employee numbers for all the employees whose manager’s employee number is 100, 101, or 201. JANET SOH JIA ER

SELECT EMPLOYEE\_ID, LAST\_NAME, SALARY, MANAGER\_ID

FROM employees

WHERE MANAGER\_ID IN (100,101,201);

1. Show all employees (id, name) earning range from 5,000 to 10,000 for the department id is 110 and sort the result from the highest to the lowest salary KUAH JIA YU

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY

FROM employees

WHERE **DEPARTMENT\_ID = 110** AND (SALARY BETWEEN 5000 AND 10000)

**ORDER BY SALARY DESC;**

1. Show all jobs (ID, title) that pay at least 15,000 salaries. LAU JUN DIAN

SELECT JOB\_ID, JOB\_TITLE

FROM jobs

WHERE MIN\_**SALARY** >= 15000;

1. List all locations with a street address that has a number “8” in it. LEE JIA JIE

SELECT LOCATION\_ID, STREET\_ADDRESS, POSTAL\_CODE, CITY, STATE\_PROVINCE, COUNTRY\_ID

FORM locations

WHERE **STREET\_ADDRESS** LIKE ‘%8%’;

1. Display the last name, job ID, and commission for all employees who are entitled to receive a commission. LEE JING JET

SELECT LAST\_NAME , JOB\_ID , COMMISSION\_PCT

FROM employees

WHERE COMMISSION\_PCT IS NOT NULL;

Multiple table queries

1. List all employees (id, name) working in the state province of California. LOW JUN YAN

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME

FROM employees E, locations L, departments D

WHERE L.LOCATION\_ID = D.LOCATION\_ID AND D.DEPARTMENT\_ID = E.DEPARTMENT\_ID AND STATE\_PROVINCE = ‘California’;

1. List employees (id, name) working in countries beginning with the letter ‘C’. NEE MEI YI

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME**, COUNTRY\_NAME**

FROM employees E, countries C, departments D, locations L

WHERE L.LOCATION\_ID = D.LOCATION\_ID AND D.DEPARTMENT\_ID = E.DEPARTMENT\_ID AND C.COUNTRY\_ID = L.COUNTRY\_ID AND COUNTRY\_NAME LIKE ‘C%’;

1. List all employees that had worked as a "SALES REPRESENTATIVE" previously (not including the current job). NEW YEE HAO

SELECT E.EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, START\_DATE,END\_DATE

FROM employees E, job\_history JH, jobs J

WHERE E.EMPLOYEE\_ID = JH.EMPLOYEE\_ID AND J.JOB\_ID = JH.JOB\_ID AND UPPER(JOB\_TITLE)= ‘SALES REPRESENTATIVE’;

1. Show all salary details (job title, salaries) for the ‘Sales’ department. NG EASON

SELECT JOB\_TITLE, MAX\_SALARY, ,MIN\_SALARY

FROM employees E, jobs J, departments D

WHERE E.JOB\_ID = J.JOB\_ID AND

D.DEPARTMENT\_ID = E.DEPARTMENT\_ID AND

DEPARTMENT\_NAME = ‘Sales’ ;

Aggregate functions

1. How many employees had been a Stock Clerk previously? ONG SHEN HO

SELECT COUNT (EMPLOYEE\_ID) AS Stock\_Clerk\_Previous

FROM jobs J, job\_history JH

WHERE JH.JOB\_ID=J.JOB\_ID AND

JOB\_TITLE=’Stock Clerk’;

1. How many current Stock Clerks are there? SEAN LOI YIT SENG

SELECT COUNT(EMPLOYEE\_ID) AS Stock\_Clerk\_Current

FROM jobs J, employees E

WHERE J.JOB\_ID=E.JOB\_ID AND JOB\_TITTLE=’Stock Clerk’;

1. Display the total number of employees who are worked in each job (include job\_id, title). TAN KANG HONG

SELECT **J.JOB\_ID, JOB\_TITLE**, COUNT(EMPLOYEE\_ID) AS Number\_of\_Employees

FROM employees E, jobs J

WHERE E.JOB\_ID = J.JOB\_ID

GROUP BY **J.JOB\_ID, JOB\_TITLE**;

1. Display the total number of employees who are worked in the **same country**, **arrange** by country id. TAN LI YUET

SELECT **C.COUNTRY\_ID, COUNTRY\_NAME**, COUNT(EMPLOYEE\_ID) AS Number\_Of\_Employees

FROM countries C, locations L, departments D, employees E

WHERE C.COUNTRY\_ID = L.COUNTRY\_ID AND L.LOCATION\_ID = D.LOCATION\_ID

AND D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

GROUP BY **C.COUNTRY\_ID,COUNTRY\_NAME**

**ORDER BY** C.COUNTRY\_ID;

1. What is the total salary of **each department** located in Canada country? WONG WENG CHENG MAVIS

SELECT **D.DEPARTMENT\_ID, DEPARTMENT\_NAME**, SUM(SALARY) AS Total\_Salary\_Of\_Each\_Department

FROM departments D, employees E, countries C, locations L

WHERE COUNTRY\_NAME = ‘Canada’ AND

C.COUNTRY\_ID = L.COUNTRY\_ID AND

L.LOCATION\_ID = D.LOCATION\_ID AND

D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

GROUP BY **D.DEPARTMENT\_ID, DEPARTMENT\_NAME**;

1. What are the departments in 'United Kingdom' that have at least 5 employees? YEW ZE XUAN

SELECT E.DEPARTMENT\_ID, DEPARTMENT\_NAME, COUNT(EMPLOYEE\_ID) AS Number\_of\_employees

FROM departments D, employees E, locations L, countries C

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID AND

D.LOCATION\_ID = L.LOCATION\_ID AND

C.COUNTRY\_ID = L.COUNTRY\_ID AND

COUNTRY\_NAME= ‘United Kingdom’

GROUP BY E.DEPARTMENT\_ID, DEPARTMENT\_NAME

HAVING COUNT(EMPLOYEE\_ID) >= 5;

1. List the job ID and total monthly salary for each job that has a total payroll exceeding 13,000. Exclude the sales representatives and sorts the list by the total monthly salary. YOON YU HONG

SELECT JOB\_ID, SUM(SALARY) AS Total\_PAYROLL

FROM jobs J, employees E

WHERE E.JOB\_ID=J.JOB\_ID AND JOB\_TILLLE= ‘ Sales Representatives’

GROUP BY JOB\_ID

HAVING SUM(salary) > 13000

ORDER BY SUM(salary); // ORDER BY Total\_PAYROLL;

Update

1. The job id of an employee 105 is changed to ‘PU\_MAN’ and the salary is increased by 5%. AARON WONG ZYI SYEN

UPDATE employees

SET SALARY=SALARY\*1.05, JOB\_ID=’PU\_MAN’

WHERE EMPLOYEE\_ID=105;

Sub-query

1. List the employee\_id, job\_id, First\_name and salary of the employees whose salary is more than the salary of Nancy (first\_name). CECILIA KONG XIN RU

SELECT EMPLOYEE\_ID,JOB\_ID,FIRST\_NAME,SALARY

FROM employees

WHERE SALARY > (SELECT SALARY

FROM employees

WHERE FIRST\_NAME = ‘Nancy’);

1. List the total number of employees in each department, whose salary is more than average salary. CHENG CAI JIE

SELECT D.DEPARTMENT\_ID,DEPARTMENT\_NAME,COUNT(EMPLOYEE\_ID)AS NUMOfEmployees

FROM employees, departments

WHERE E.DEPARTMENT\_ID = D.DEPARTMENT\_ID AND SALARY > (SELECT AVG(SALARY) FROM employees)

GROUP BY D.DEPARTMENT\_ID, DEPARTMENT\_NAME;

1. Remove all employees who worked at the department which has the word ‘ne’ in department name. CHI PUI MUN

DELETE FROM employees

WHERE DEPARTMENT\_ID IN (SELECT DEPARTMENT\_ID

FROM department

WHERE DEPARTMENT\_NAME LIKE ‘%ne%’) ;

1. Increase employee’s salary as 5% who’s worked as a ‘Programmer’ (job title). CHUAH SHEE YEAP

UPDATE employees

SET SALARY = SALARY \* 1.05

WHERE JOB\_ID = (SELECT JOB\_ID

FROM jobs

WHERE JOB\_TITLE = ‘Programmer’);

## Tutorial 4: Data Models

## Question 1

Describe any TWO (2) features of conceptual data model in database design.

Give an example of diagram to produce the conceptual data model.

GOH KHAI JUN

A conceptual data model identifies the highest-level relationships between the different entities. Features of conceptual data model include:

* Includes the important entities and the relationships among them.
* No attribute is specified
* No primary key is specified

The examples of a diagram to produce the conceptual data model are Entity-Relationship Model(ERD) and Object-oriented data model (UML).

## Question 2

Explain TWO (2) main purposes of business rules in designing a database.

HAR CHUN WAI

* **They also allow the creators to understand business processes, and the nature, role and scope of the data.**
* **They are a communication tool between users and creators, and they also help standardize the company’s view of the data.**
* They are used for the organization that stores or uses data to be an explanation of policy, procedure or principle.
* They are used to promote creation of an accurate data model.

## Question 3

Construct an ERD using Crow’s Foot notation based on sample given:

Given sample record as below:

Each sales representative issues at least one invoice to a customer. Each invoice may listed many ordered items. Each customer may receive many invoices at different date issued.

HO JING XIAN

## 

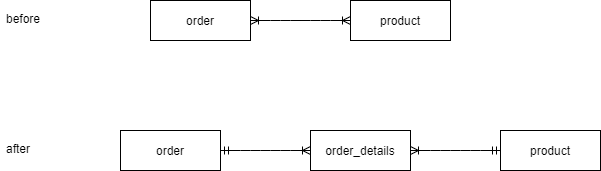
## Question 4

## How to resolve many-to-many relationships between the 2 entities? Give an example for illustration.

## JANET SOH JIA ER

-Use associative entities to resolve many-to-many relationships between the 2 entities and to form a new 2 pairs of one-to-many relationships.

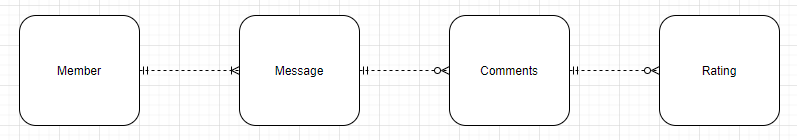
-Example: Each order can consists of one or more products, and each product can be placed by one or more orders. In order to solve many-to-many relationships, the order\_details entity (associative entity) will appear between order entity and product entity.



## Question 5

A newly establish social media company has requested you to design database to model memberships, message posted by members and comments collected.

The company would like to record member’s information such as their id, name, phone number, and email address. Each member is allowed to post at least 1 message. Information captured from members’ posted messages included id, posted title, date of the message is posted, privacy setting such as private or public. Each message posted may or may not receive comments. However, each time a comment is received, information about when the comment was posted, the total posted comment, commentID, and the content message will be recorded too. Each posted comment will be given a rating (rate 1 to 5) or none.

a) Draw an Entity-Relationship Diagram (ERD) for the above scenario using the *Crow’s Foot notation* (exclude all attributes). Resolve many-to-many relationships, if any. 

b) For each of the entities in the ERD drawn in part a), list all relevant attributes using Database Design Language (DBDL) format. In your listing, show all the primary keys and foreign keys (if any) clearly. Underline all the primary keys and identify the foreign keys with an \*.

KUAH JIA YU

MEMBER (Member\_ID, Member\_Name, Phone\_Number, Email)

MESSAGE (Message\_ID, Title, Date, PrivacySetting, Member\_ID\*)

COMMENT (Comment\_ID, Date\_Comment, Content, Number\_Posted, Message\_ID\*)

RATING (RatingNo, Rating, Comment\_ID\*)

- \* put on many side

## Question 6

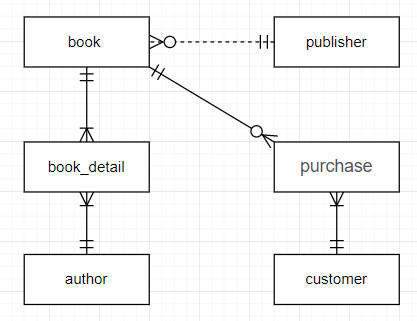
Vitech Popular Bookstores (VPB), a chain of bookstores wants to keep track of the books it sells, their publishers(pearsoned, wiley), their authors, and the customers who buy them.

The BOOK entity has 4 attributes (Book Number(identifier), Name, Publication Year, Pages). A book has exactly one publisher. A book must have at least one author but can have many. A book that VPB carries may not as yet have been bought by any of its customers or may have been bought by many of its customers.

PUBLISHER has 5 attributes (Publisher Name, City, Country, Telephone, Year Founded); Publisher Name is the unique identifier of the PUBLISHER entity. A publisher may have published many books that VPB carries; however, VPB also wants to be able to keep track of some publishers that currently have no books in VPB's inventory.

For a person to be of interest to VPB as an author, she must have written at least one and possibly many books that VPB carries. An AUTHOR has 4 attributes (Author Number, Name, Year Born, Year Died); Author Number uniquely identifies an author.

For a customer to be of interest to VPB, he must have bought at least one book and possibly many. CUSTOMER has 6 attributes (Customer Number, Name, Street, City, State, and Country) with Customer Number as the unique identifier.

a) Draw an Entity-Relationship Diagram (ERD) for the above scenario using the *Crow’s Foot notation* (exclude all attributes). Resolve many-to-many relationships, if any. 

b) For each of the entities in the ERD drawn in part a), list all relevant attributes using Database Design Language (DBDL) format. In your listing, show all the primary keys and foreign keys (if any) clearly. Underline all the primary keys or composite keys and identify the foreign keys with an \*.

LAU JUN DIAN

BOOK (Book\_ID, Book\_Name, Book\_Publication\_Year, Book\_Pages,Publisher\_Name\*)

PUBLISHER (Publisher\_Name, Publisher\_City, Publisher\_Country, Publisher\_Telephone, Publisher\_Year\_Founded)

AUTHOR (Author\_ID, Author\_Name, Author\_Year\_Born, Author\_Year\_Died)

CUSTOMER (Customer\_ID, Customer\_Name, Customer\_Street, Customer\_City, Customer\_State, Customer\_Country)

BOOK\_DETAIL(Author\_ID\*,Book\_ID\*)

PURCHASE(Customer\_ID\*,Book\_ID\*,Purchase\_Price)

## Tutorial 5: Normalization

For all questions in part a):

Normalize the table to a set of Third Normal Form (3NF) relations. Your answer should show all the 3 stages of normalization (1NF, 2NFand 3NF) by using Database Definition Language (DBDL) format (underline all the primary key(s), composite key(s) and use an \* to indicate the foreign keys). State the component that is or are removed from each Normal Form.

* + 1. Dream Sdn Bhd is a tour agency is using traditional file-based system to manage and maintain all records of their customers’ bookings. TourPackages Table shown below is an example of data file stored in their system.

Sample Tour Package Records

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Package Code** | **Package Desc** | **Price** | **Tour Guide** | **TourGuide HP** | **CustNo** | **CustName** | **CustHP** | **DepartDate** |
| B123 | 8 days Bangkok | 4,800 | Andy | 011-1234567 | Y88 | YeohLW | 016-9871234 | 02-02-2018 |
| L99 | LeeQW | 012-6547890 |
| Y10 | Yumiko | 011-76589432 |
| R22 | Rina | 010-97531246 |
| T007 | 10 days Turkey | 10,800 | Sally | 012-9876543 | S12 | Susan | 018-98712347 | 08-03-2018 |
| E236 | 5 days England | 5,800 | Ryan | 016-6520888 | A98 | Andrew | 012-12347890 | 11-04-2018 |
| Z80 | Zani | 016-54632100 |
| J560 | 5 days Japan | 6,800 | Yuki | 017-3850088 | K34 | Kumichi | 011-91743002 | 09-05-2018 |
| P66 | Pumar | 019-7890088 |
| N68 | Natalie | 010-8822569 |

(a) Normalize the table given to a set of third normal form (3NF) relations using DBDL format.

**1NF(Remove Repeating Group)**

TourPackage(PackageCode, PackageDesc, Price, TourGuide, TourGuideHP, CustNo, CustName, CustHP, DepartDate)

|

Package(PackageCode, PackageDesc, Price, TourGuide, TourGuideHP, DepartDate)

TourPackage(PackageCode\*, CustNo, CustName, CustHP)

**2NF(Remove Partial Dependency)**

Package(PackageCode, PackageDesc, Price, TourGuide, TourGuideHP, DepartDate) - 2 objects

TourPackage(PackageCode\*, CustNo\*)

Customer(CustNo,CustName,CustHP)

**3NF(Remove Transitive Dependency)**

Package(PackageCode,PackageDesc,Price,TourGuide\*,DepartDate) A-> B

TourGuide (TourGuide,TourGuideHP) B-> C

Customer(CustNo,CustName,CustHp)

TourPackage(PackageCode\*,CustNo\*)

(b) Based on the sample data given in the table, discuss each of the following data anomalies with a specific *example:*

(i) Insertion anomaly

It is not possible to add a new customer unless that new customer is registered for a tour package.

//

It is not possible to add a new tour package unless that new package is registered by the customer.

(ii) Modification anomaly

When we update the Package Description (E326) from “5 days England” to “7 days England”, we have to update the similar package description in another row, otherwise it will cause inconsistency of data.

(iii) Deletion anomaly

When we delete the package (T007) record, we have to delete the Susan details (S12).

LEE JIA JIE

## 2. Refer to the table below:

Normalize the table given below to a set of third normal form (3NF) relations using DBDL format.

Table shows the Student\_Subject records:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Student**  **Id** | **Student Name** | **Tutorial**  **Class** | **Subject**  **Code** | **Subject**  **Title** | **Credit**  **Hour** | **Tuition Fees** | **Grade** |
| A001023 | Bredly | M45 | AACS2164 | Database | 4 | 280.00 | B |
| A001023 | Bredly | M45 | AACS2184 | System Design | 3 | 210.00 | B |
| A123450 | Ai Li | M23 | AAMS1234 | Mathematics | 5 | 350.00 | C |
| A123450 | AI Li | M23 | AACS2164 | Database | 4 | 280.00 | A |
| A112234 | Branda | M40 | AACS2164 | Database | 4 | 280.00 | B |
| A009785 | William | M44 | AACS2184 | System Design | 3 | 210.00 | C |

Hint: The tuition fees are calculated based on the credit hour.

(a) Normalize the table given to a set of third normal form (3NF) relations using DBDL format.

**1NF(Remove Repeating Group)**

StudentSubject(StudentId ,StudentName, TutorialClass, SubjectCode, SubjectTitle, CreditHour, TuitionFees, Grade)

|

Student(StudentId ,StudentName,TutorialClass)

StudentSubject(StudentId\*, SubjectCode,SubjectTitle,CreditHour,TuitionFees,Grade)

**2NF(Remove Partial Dependency)**

Student(StudentId, StudentName, TutorialClass)

StudentSubject(StudentId\*, SubjectCode\*, Grade)

SubjectDetails(SubjectCode, SubjectTitle,CreditHour, TuitionFees) - 2 objects

**3NF(Remove Transitive Dependency)**

Student(StudentId, StudentName, TutorialClass)

Student\_Subject(StudentId\*, SubjectCode\*, Grade)

TuitionCharge(CreditHour ,TuitionFees) B-> C

Hint: The tuition fees are calculated based on the credit hour.

SubjectDetails(SubjectCode ,SubjectTitle, CreditHour\*) A-> B

(b) Based on the sample data given in the table, discuss each of the following data anomalies with a specific *example:*

(i) Insertion anomaly

It is not possible to add the new Student unless that new Student is registered for a Subject.

(ii) Modification anomaly

When we update subject title of AACS2164 from Database to Advance Database,we also need to update the similar subject title in another row, if not it will cause data inconsistency.

(iii) Deletion anomaly

When we delete the student named Ai Li (A123450) record, we also will delete the subject detail: Mathematics (AAMS1234).

LEE JING JET

3. The table shown below contains records about patients, health care providers, patients’ visits to a clinic, and diagnoses made by health care providers. **(LOW JUN YAN)**

HealthCare records:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **VisitNo** | **VisitDate** | **PatNo** | **PatAge** | **PatCity** | **PatZip** | **ProvNo** | **ProvSpecialty** | **Diagnosis** |
| V20030 | 13/1/2005 | P1 | 35 | Lahat | 31600 | D1 | Internist | Ear Infection |
| V20030 | 13/1/2005 | P1 | 35 | Lahat | 31600 | D2 | Nurse Practitioner | Influenza |
| V82020 | 20/1/2005 | P3 | 20 | Tronoh | 32200 | D2 | Nurse Practitioner | Pregnancy |
| V73220 | 18/1/2005 | P2 | 62 | Taiping | 34500 | D3 | Cardiologist | Murmur |

Hint: The zip code of the patient can be used to determine the city.

(a) Normalize the table given to a set of third normal form (3NF) relations using DBDL format.

**1NF(REMOVE REPEATING GROUP)**

HealthCare(VisitNo, VisitDate, PatNo, PatAge, PatCity, PatZip, ProvNo, ProvSpecialty, Diagnosis)

|

Visit(VisitNo, VisitDate, PatNo, PatAge, PatCity, PatZip)

HealthCare(VisitNo\*, ProvNo, ProvSpecialty, Diagnosis)

**2NF(REMOVE PARTIAL DEPENDENCIES)**

Visit(VisitNo, VisitDate, PatNo, PatAge, PatCity, PatZip) - 2 objects

Provider(ProvNo, ProvSpecialty)

HealthCare(VisitNo\*, ProvNo\*, Diagnosis)

**3NF(REMOVE TRANSITIVE DEPENDENCIES)**

Visit(VisitNo, VisitDate, PatNo\*) A1 -> B1

Patient (PatNo, PatAge, PatZip\*) B1 -> C1 , A2-> B2

Zip(PatZip, PatCity) B2 -> C2

Hint: The zip code of the patient can be used to determine the city.

HealthCare(VisitNo\*, ProvNo\*, Diagnosis)

Provider(ProvNo, ProvSpecialty)

(b)Based on the sample data given in the table, discuss each of the following data anomalies with a specific *example:*

(i) Insertion anomaly (Inability to add new records without adding redundant data.)

It is not possible to add the new provider unless that new provider serves a patient on visit.

It is not possible to add the new patient unless the new patient is served by any providers.

(ii) Modification anomaly (Inability to accurately and efficiently maintain data.)

When we update the Provider Specialty (D2) from Nurse Practitioner to Nurse Professional,

we need also update the similar Provider Specialty in another row, if not it will cause data

inconsistency.

(iii) Deletion anomaly (Inability to delete unwanted data without deleting data that you need to retain.)

When we delete the Visit (V73220) record we also will delete the provider details:

Cardiologist (D3).

4. Given a Book\_Order table as shown below, normalize it to a set of 3NF relations in DBDL.

Book Order Records

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| BookID | BookTitle | BookCategory | PublisherID | PublisherName | CourseID | CourseName | OrderCopy |
| B001 | Database | Computer | P101 | Wiley | O111 | Oracle | 95 |
| M222 | MySQL | 55 |
| B002 | Programming | Computer | P101 | Wiley | J888 | Java Program | 80 |
| C777 | C++ program | 60 |
| B003 | Business Accounting | Business | P102 | Prentice | A211 | Accounting | 90 |
| B004 | Business Law | Business | P103 | Pearsoned | A211 | Accounting | 66 |

(a) Normalize the table given to a set of third normal form (3NF) relations using DBDL format.

**1NF(Remove Repeating Group)**

Book\_Order(BookID, BookTitle, BookCategory, PublisherID, PublisherName, CourseID, CourseName, OrderCopy)

|

Book\_Order(BookID, BookTitle, BookCategory, PublisherID, PublisherName)

Book\_Course(BookID\*, CourseID\*, CourseName, OrderCopy)

**2NF(Remove Partial Dependencies)**

Book\_Order(BookID, BookTitle, BookCategory, PublisherID, PublisherName) - 2 objects

Course(CourseID, CourseName)

Book\_Course(BookID\*, CourseID\*, OrderCopy)

**3NF(Remove Transitive Dependencies)**

Book\_Order(BookID, BookTitle, BookCategory, PublisherID\*)

Publisher(PublisherID, PublisherName)

Course(CourseID, CourseName)

Book\_Course(BookID\*, CourseID\*, OrderCopy)

(b) Based on the sample data given in the table, discuss each of the following data anomalies with a specific *example:*

(i) Insertion anomaly

It is not possible to add the new course unless the new course is ordered the book.

(ii) Modification anomaly

When we update the course name (A211) from Accounting to Financial Accounting,

we need also update the similar course name in another row, if not it will cause data

inconsistency.

(iii) Deletion anomaly

When we delete the book title named programming (B002) record we also will delete the

course details: Java Program(J888) and C++ program(C777).

NEE MEI YI

5. A sample of Delivery details is shown as below:

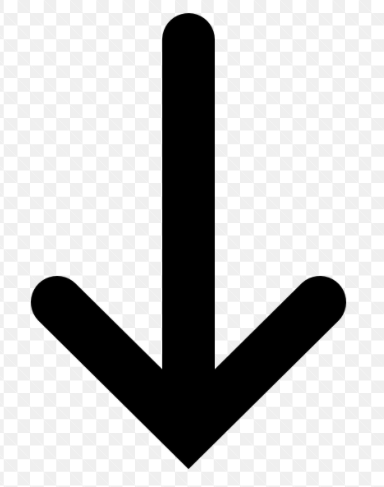
Delivery details records:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Delivery Order No | Delivery Order Date | CustID | HpNo | CustName | Address | Item  Code | Description | Qty Delivered |
| DO001 | 12-05-19 | AMD12 | 012587895 | AMD Sdn Bhd | 23 Taman Suria | CO98 | Phone case | 50 |
| AB56 | Armband | 30 |
| DP12 | Dust plug | 5 |
| DO002 | 12-05-19 | HP2X5 | 011789456 | LTK Sdn Bhd | 5 Willis Ave | PP65 | Pouch | 25 |
| CO98 | Phone case | 20 |
| DO003 | 03-06-19 | CK134 | 016963258 | CK Enterprise | Sun Avn. | CP57 | Cooler pad | 10 |
| DO004 | 12-08-19 | AMD12 | 012587895 | AMD Sdn Bhd | 23 Taman Suria | CC23 | Car charger | 5 |
| PP65 | Pouch | 15 |
| CO98 | Phone case | 10 |

(a) Normalize the table given to a set of third normal form (3NF) relations using DBDL format.

**1NF(Remove Repeating Group)**

DeliveryDetails(DeliveryOrderNo, DeliveryOrderDate, CustID, HpNo, CustName, Address, ItemCode, Description, QtyDelivered)

****

DeliveryDetails(DeliveryOrderNo, DeliveryOrderDate, CustID, HpNo, CustName, Address)

DeliveryItem(DeliveryOrderNo\*, ItemCode\*, Description, QtyDelivered)

**2NF(Remove Partial Dependency)**

Delivery(DeliveryOrderNo, DeliveryOrderDate, CustID, HpNo, CustName, Address) - 2 objects

DeliveryDetails(DeliveryOrderNo\*, ItemCode\*, QtyDelivered)

Item(ItemCode, Description)

**3NF(Remove Transitive Dependency)**

Delivery(DeliveryOrderNo\*, DeliveryOrderDate, CustID\*) A-> B

DeliveryDetails(DeliveryOrderNo\*, ItemCode\*, QtyDelivered)

Customer(CustID, HpNo, CustName, Address) B-> C

Item(ItemCode, Description)

(b) Based on the sample data given in the table, discuss each of the following data anomalies with a specific *example:*

(i) Insertion anomaly

It is not possible to add a new Item unless that new Item is ordered by a customer.

(ii) Modification anomaly

When we update the Item Description for Item Code (CO98) from Phone case to Phone cover, we have to update the similar item description in another row, otherwise it will cause inconsistency of data.

(iii) Deletion anomaly

When we delete the delivery order (DO003) record, we have to delete the item details: Cooler pad (CP57).

NEW YEE HAO

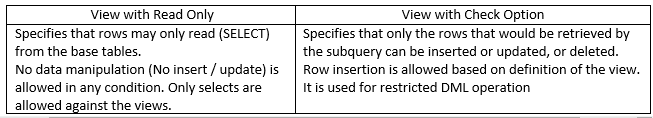
**Tutorial 6: Advanced SQL**

1. What’s the purpose of creating a view with *read only* and *with check* options?

NG EASON

The purpose of creating a view with read only is to ensure that no DML operations can be performed on this view.

The purpose of creating a view with check options is to ensure that DML operations can be performed on this view which must match all the conditions specify on where-clause.



*Refer to the Tutorial 3: HR ERD,* **Write SQL queries** to do the following:

1. Create a view that will display the following output when queried.

LOCATION COUNTRY REGION

---------------------------------------- --------- ---------------- -------- -------------

1200 Tokyo Prefecture Japan Asia

1400 Texas United States of America Americas

1500 California United States of America Americas

1600 New Jersey United States of America Americas

1700 Washington United States of America Americas

1800 Ontario Canada Americas

1900 Yukon Canada Americas

2100 Maharashtra India Asia

2200 New South Wales Australia Asia

2500 Oxford United Kingdom Europe

2600 Manchester United Kingdom Europe

where the 'LOCATION' column is **location\_id and state\_province** combined;

'COUNTRY' column is the name of a country and

'REGION' column is the name of a region

Another requirement is do not include records that do not have state\_province name.

ONG SHEN HOI

CREATE OR REPLACE VIEW LocationRegion (LOCATION, COUNTRY, REGION) AS

SELECT LOCATION\_ID || ‘ ‘ || STATE\_PROVINCE, COUNTRY\_NAME, REGION\_NAME

FROM locations L, countries C, regions R

WHERE L.COUNTRY\_ID=C.COUNTRY\_ID AND C.REGION\_ID = R.REGION\_ID AND STATE\_PROVINCE IS NOT NULL: (optional)

Multiple table queries (from tutorial 3, Q8-Q11 (3,4,5 & 6))

Use the ‘**Joins with the ON Clause’** to answer the following.

1. List all employees (id, name) working in the state province of California. SEAN LOI YIT SENG

SELECT E.EMPLOYEE\_ID, E.FIRST\_NAME, E.LAST\_NAME

FROM employees E JOIN departments D ON E.DEPARTMENT\_ID = D.DEPARTMENT\_ID JOIN Locations L ON D.LOCATION\_ID = L.LOCATION\_ID

WHERE STATE\_PROVINCE = ‘California’;

1. List employees (id, name) working in countries beginning with the letter ‘C’. TAN KANG HONG

SELECT E.EMPLOYEE\_ID, E.FIRST\_NAME, E.LAST\_NAME**,** C.COUNTRY\_NAME

FROM employees E JOIN departments D ON D.DEPARTMENT\_ID = E.DEPARTMENT\_ID JOIN locations L ON L.LOCATION\_ID = D.LOCATION\_ID

JOIN countries C ON C.COUNTRY\_ID = L.COUNTRY\_ID AND COUNTRY\_NAME LIKE ‘C%’;

1. List all employees that had worked as a "SALES REPRESENTATIVE" previously (not including the current job). TAN LI YUET

SELECT E.\*

FROM employees E JOIN job\_history JH ON E.EMPLOYEE\_ID = JH.EMPLOYEE\_ID JOIN Jobs J ON J.JOB\_ID = JH.JOBS\_ID

WHERE UPPER(JOB\_TITLE) = ‘SALES REPRESENTATIVE’;

1. Show the salary details (job title, salaries) for the IT department. WONG WENG CHENG MAVIS

SELECT DISTINCT JOB\_TITLE, MAX SALARY, MIN SALARY

FROM employees E JOIN jobs J ON E.JOB\_ID = J.JOB\_ID JOIN departments D ON D.DEPARTMENT\_ID = E.DEPARTMENT\_ID

WHERE DEPARTMENT\_NAME = ‘IT’;

1. Display all employees who are worked in the same country. YEW ZE XUAN

SELECT C.COUNTRY\_ID, C.COUNTRY\_NAME, E.EMPLOYEE\_ID, E.FIRST\_NAME

FROM employees E JOIN departments D

ON E.DEPARTMENTS\_ID = D.EMPLOYEES\_ID

JOIN locations L

ON D.LOCATION\_ID = L.LOCATION\_ID

JOIN countries C

ON L.COUNTRY\_ID = C.COUNTRY\_ID;

**OUTER JOINS**

1. Based on the following records in Staff and Project tables, produce a resulting table after each of the following relational **Join** operations has been performed:

|  |  |
| --- | --- |
| **LEFT** | **RIGHT** |
| Staff   |  |  |  | | --- | --- | --- | | Staff\_Code | Staff\_Name | Position | | M01 | John Tan | Manager | | E01 | Ahmad | Executive | | S01 | Gopal | Supervisor | | Project   |  |  | | --- | --- | | Project\_No | Staff\_Code | | 1010 | M01 | | 1011 | E01 | | 1012 | E01 | | 1013 | C01 | |

-Staff **Left Outer Join** Project

-Staff **Right Outer Join** Project

-Staff **Full Outer Join** Project

YOON YU HONG

-Staff **Left Outer Join** Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Staff\_Code | Staff\_Name | Position | Project\_No | Staff\_Code |
| M01 | John Tan | Manager | 1010 | M01 |
| E01 | Ahmad | Executive | 1011 | E01 |
| E01 | Ahmad | Executive | 1012 | E01 |
| S01 | Gopal | Supervisor |  |  |

-Staff **Right Outer Join** Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Staff\_Code | Staff\_Name | Position | Project\_No | Staff\_Code |
| M01 | John Tan | Manager | 1010 | M01 |
| E01 | Ahmad | Executive | 1011 | E01 |
| E01 | Ahmad | Executive | 1012 | E01 |
|  |  |  | 1013 | C01 |

-Staff **Full Outer Join** Project (LEFT + RIGHT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Staff\_Code | Staff\_Name | Position | Project\_No | Staff\_Code |
| M01 | John Tan | Manager | 1010 | M01 |
| E01 | Ahmad | Executive | 1011 | E01 |
| E01 | Ahmad | Executive | 1012 | E01 |
| S01 | Gopal | Supervisor |  |  |
|  |  |  | 1013 | C01 |

**SET Operators**

1. Based on the following records in StaffA and StaffB tables, produce a resulting table after each of the following relational **Set** operations has been performed:

|  |  |
| --- | --- |
| **StaffA** | **StaffB** |
| |  |  |  | | --- | --- | --- | | Staff\_Code | Staff\_Name | Position | | M01 | John Tan | Manager | | E01 | Ahmad | Executive | | S01 | Gopal | Supervisor | | |  |  |  | | --- | --- | --- | | Staff\_Code | Staff\_Name | Position | | D01 | Stephen | General Manager | | E01 | Ahmad | Executive | | G01 | Julia | Clerk | | S01 | Gopal | Supervisor | |

-StaffA **Union** StaffB

-StaffA **Union All** StaffB

-StaffA **Intersect** StaffB

-StaffA **Minus** StaffB

AARON WONG ZYI SYEN

-StaffA **Union**  StaffB

|  |  |  |
| --- | --- | --- |
| **Staff\_Code** | **Staff\_Name** | **Position** |
| M01 | John Tan | Manager |
| E01 | Ahmad | Executive |
| S01 | Gopal | Supervisor |
| D01 | Stephen | General Manager |
| G01 | Julia | Clerk |

-StaffA **Union All** StaffB

|  |  |  |
| --- | --- | --- |
| **Staff\_Code** | **Staff\_Name** | **Position** |
| M01 | John Tan | Manager |
| E01 | Ahmad | Executive |
| S01 | Gopal | Supervisor |
| D01 | Stephen | General Manager |
| E01 | Ahmad | Executive |
| G01 | Julia | Clerk |
| S01 | Gopal | Supervisor |

-StaffA **Intersect** StaffB

|  |  |  |
| --- | --- | --- |
| **Staff\_Code** | **Staff\_Name** | **Position** |
| E01 | Ahmad | Executive |
| S01 | Gopal | Supervisor |

-StaffA **Minus** StaffB

|  |  |  |
| --- | --- | --- |
| **Staff\_Code** | **Staff\_Name** | **Position** |
| M01 | John Tan | Manager |

**Tutorial 7: Database Administration and Security**

1. How would you ensure data availability of system? CECILIA KONG XIN RU

Hardware failures - provide redundancy for fault tolerance

Loss of data - Implement database mirroring,in order to cover the loss data

Human error - Provide standard operating procedures,training,documentation ,to reduce the chance of human errors.

Maintenance downtime - automated and non-disruptive maintenance utilities to avoid system breakdown suddenly.

Network problems -careful monitor the network,firewall and router traffic to reduce the chance of the system being invaded by the outsiders.

1. Recommend any **TWO (2)** features of database software security for online banking service. CHENG CAI JIE

-**View or subschemas**

- only allow the user to access the part of the database that is related to the use

**-Authorization rule**s

- allow user control of insert, update, delete or all privileges on the database record

**-User-defined procedure**

- follow the step to access the database records.

-**Encryption**

-encode and decode the original message

**-Authentication schemes**

- verification of human-to-machine of credentials required for confirmation user authenticity

1. Briefly describe the **FOUR (4)** DBMS facilities that are required for database backup and recovery. CHI PUI MUN

* **Backup Facilities**

provide periodic backup copies of portions of the entire database or subsets.

* **Journalizing Facilities**

maintain an audit trail transactions and database changes

* **Checkpoint Facilities**

allow periodic suspension of all processing and synchronization of a database’s file and journals.

* **Recovery manager**

allows DBMS to restore the database to a correct condition and restart processing transactions.

1. Suggest **ONE (1)** suitable recovery technique for database destruction of online registration system CHUAH SHEE YEAP

