

TASK A

• Construct a conceptual and logical model for the following case studies:

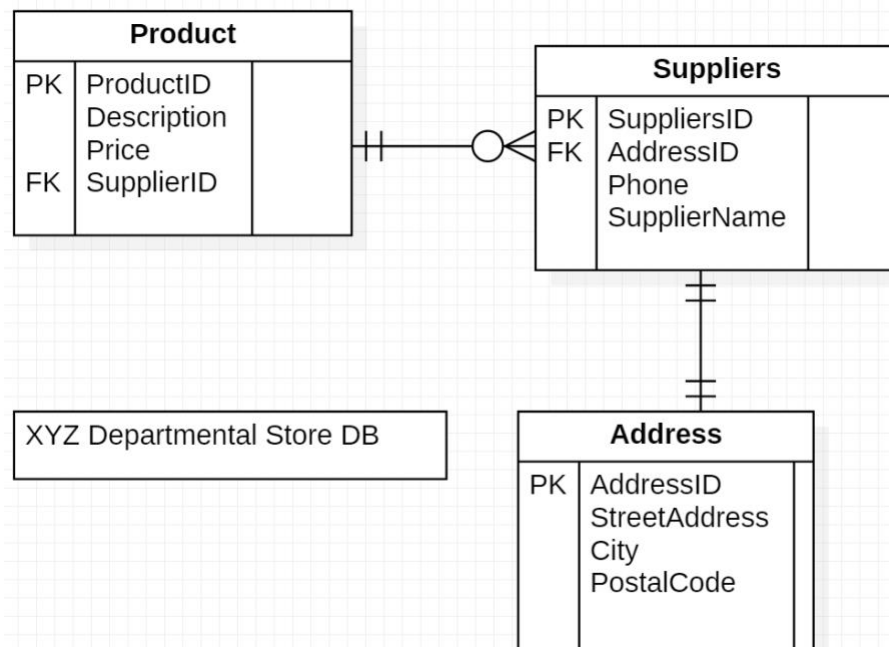
1. XYZ departmental store wants to represent information about their products in a database. Each product has a description, a price and a supplier. Suppliers have addresses, phone numbers, and names. Each address is made up of a street address, a city, and a postcode.

2. ABC University is a large institution with several campuses. Each campus has a different name, address, distance to the city center and the only bus running to the campus. Each campus has one club. The name of the club, the building in which the club is located, the phone number of the club and the multiple sports which club offers, should all be recorded. The University consists of several faculties, such as the Art Faculty, the Science Faculty, and so on. Each faculty has a name, dean and building. A faculty may be divided into several schools, for example, the Science Faculty has a School of Physics and a School of Chemistry. Each school belongs to one faculty only and is located on just one campus, but one campus maybe the location of many schools. Every school has name and a building assigned to. Each school offers different programmes and each programme can be offered by only one school. Each programme has a unique code, title, level and duration. Each programme comprises several courses, different programmes have different courses. Each course has a unique code and course title. Some courses may have one or more prerequisite courses and one course can be the prerequisite course of some other courses. Each of the students is enrolled in a single programme of study which involves a fixed core of courses specific to that programme as well as a number of electives taken from other programmes. Students work on courses and are awarded a grade in any course if he/she passes the course. Otherwise, the student has to re-take the failed course. The system needs to record the year and term in which the course was taken, and the grade awarded to the student. Every student has a unique ID. The system also keeps the student name, birthday and the year he/she enrolled in the course. The school employs lecturers to teach the students. A lecturer can work for one school only. Each lecturer is assigned an ID which is unique across the whole university. The system keeps the lecturer's name, title and the office room. A supervisor maybe in charge of several lecturers, but a lecturer, however, reports to only one supervisor. A lecturer can teach many different courses. A course may also have been taught by many different lecturers. The university is operated by committees. Each faculty has to have several committees with the same titles across the university, such as the Faculty Executive, the Post Graduate Studies Committee, the Health and Sanitary Committee, and so on. The committees meet regularly, such as weekly or monthly.

Solutions:

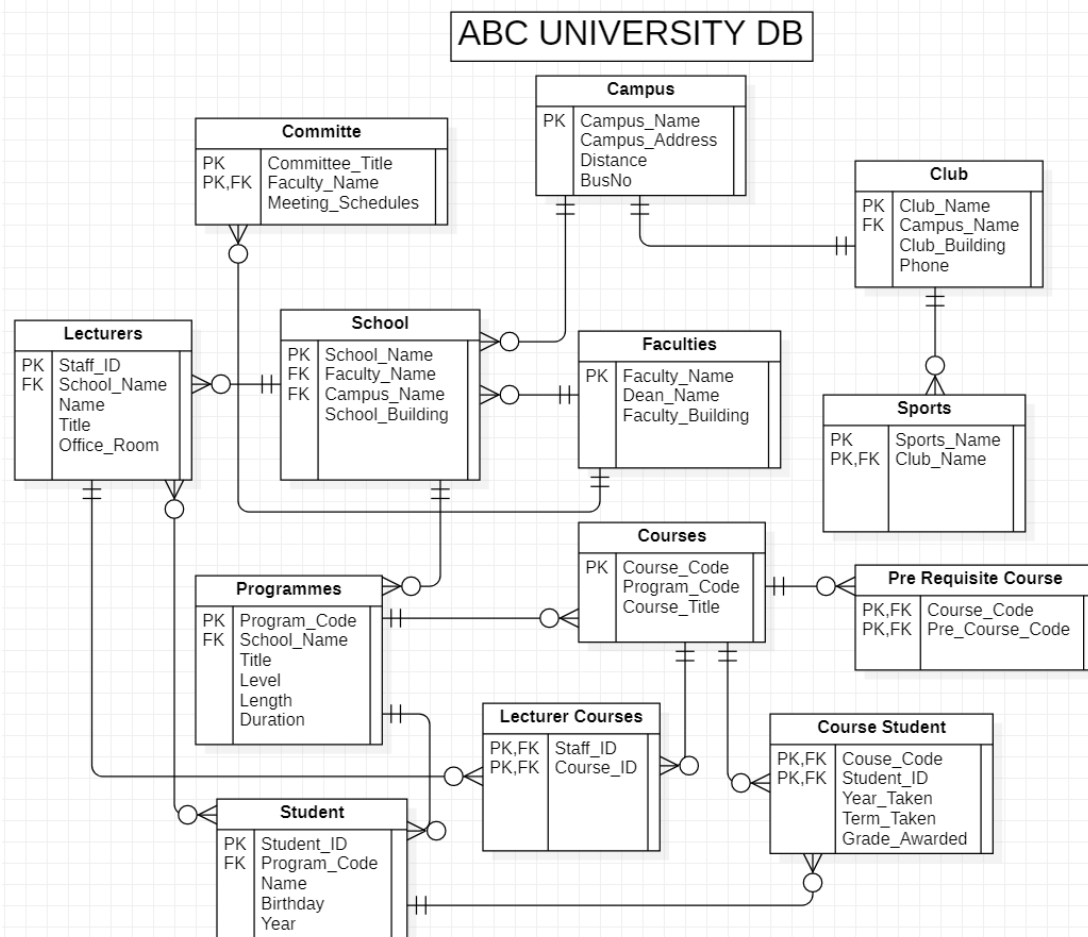
Task A

XYZ departmental store wants to represent information about their products in a database. Each **product** has a **description**, a **price**, and a **supplier**. **Suppliers** have **addresses**, **phone numbers**, and **names**. Each **address** is made up of a **street address**, a **city**, and a **postcode**.



Lec 3 - 4 (Data & Conceptual Modeling)

ABC University is a large institution with several campuses. Each **campus** has a different **name**, **address**, **distance** to the city center and the only **bus** running to the campus. Each **campus** has one **club**. The **name** of the club, the **building** in which the club is located, the **phone** number of the club and the multiple **sports** which club offers, should all be recorded. The University consists of several **faculties**, such as the Art Faculty, the Science Faculty, and so on. Each faculty has a **name**, **dean** and **building**. A faculty may be divided into several **schools**, for example, the Science Faculty has a School of Physics and a School of Chemistry. Each school belongs to one **faculty** only and is located on just one **campus**, but one campus maybe the location of many schools. Every school has **name** and a **building** assigned to. Each school offers different **programmes** and each programme can be offered by only **one** school. Each programme has a **unique code**, **title**, **level** and **duration**. Each programme comprises several **courses**, different programmes have different courses. Each **course** has a unique **code** and **course title**. Some courses may have one or more **prerequisite** courses and one course can be the prerequisite course of some other **courses**. Each of the **students** is enrolled in a single programme of study which involves a fixed core of **courses** specific to that programme as well as a number of electives taken from other programmes. Students work on courses and are awarded a **grade** in any course if he/she passes the course. Otherwise, the student has to re-take the failed course. The system needs to record the **year** and **term** in which the course was taken, and the **grade** awarded to the student. Every student has a **unique ID**. The system also keeps the student **name**, **birthday** and the **year** he/she enrolled in the course. The school employs **lecturers** to teach the students. A lecturer can work for **one** school only. Each lecturer is assigned an **ID** which is **unique** across the whole university. The system keeps the lecturer's **name**, **title** and the **office room**. A supervisor maybe in charge of several lecturers, but a lecturer, however, reports to only one supervisor. A lecturer can teach many different **courses**. A course may also have been taught by **many** different lecturers. The university is operated by **committees**. Each faculty has to have several committees with the same **titles** across the university, such as the Faculty Executive, the Post Graduate Studies Committee, the Health and Sanity Committee, and so on. The committees meet regularly, such as weekly or monthly.



TASK A

Student_Id	College_Id	Student_Name	Rtu_Roll_No	Father_Name	Address	Branch_Id	Session
1	1210038	Pankaj	12emccs138	Suresh	Alwar	1	2012-16
2	1210039	Priya	12emccs139	Ram	Mathura	1	2012-16
3	1310048	Rahul	13emmce210	Sambhu	Alwar	3	2013-17
4	1310078	Neeraj	13emcve078	Nathu	Jaipur	5	2013-17
5	1210047	Sandeep	12emccs047	Om Parkash	Alwar	1	2012-16
6	1210048	Sanjeev	12emece048	Omi	Delhi	2	2012-16

SUPER KEY ?

CANDIDATE KEY ?

PRIMARY KEY ?

FOREIGN KEY ?

Solutions:

Super Key: Student_Id, College_Id, Student_Name, Rtu_Roll_No, Father_Name, Address.

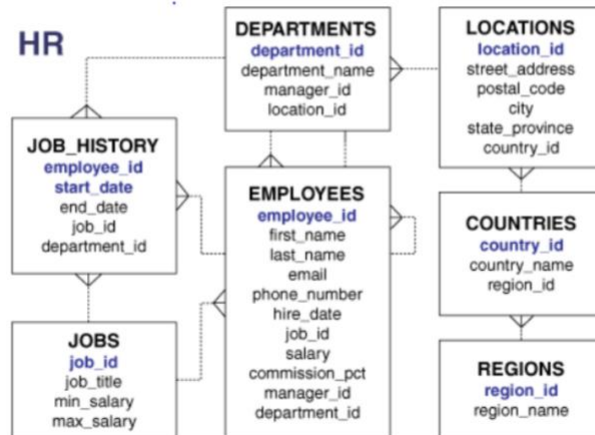
Candidate Key: Student_Id, Student_Name + Father_Name, Rtu_Roll_No.

Primary Key: Student_Id.

Foreign Key: No Foreign Key.

TASK A

1. Create the tables to implement the following database model.



TASK B

1. Create a table named Test_1 to in accordance with the below mentioned requirements.

Field Name	Data Type	Size	Decimal Places	NULL
ord_num	decimal	6		No
ord_amount	decimal	12	2	Yes
ord_date	date			No
cust_code	char	6		No
agent_code	char	6		No

Solutions:

Task A

```
CREATE TABLE job_history (  
    employee_id INT,  
    start_date DATE,  
    end_date DATE,  
    job_id INT UNIQUE,  
    department_id INT,  
  
    PRIMARY KEY(employee_id, start_date, job_id)  
  
);  
  
CREATE TABLE jobs (  
    job_id INT,  
    job_title VARCHAR(20),  
    min_salary INT,  
    max_salary INT,  
  
    PRIMARY KEY(job_id),  
    FOREIGN KEY(job_id) REFERENCES job_history(job_id)  
);  
  
CREATE TABLE employees (  
    employee_id INT,  
    first_name VARCHAR(20),  
    last_name VARCHAR(20),  
    email VARCHAR(50),  
    phone_number INT,  
    hire_date DATE,  
    job_id INT,  
    salary INT,  
    commission_pct INT,  
    manager_id INT UNIQUE,  
    department_id INT,  
  
    PRIMARY KEY(employee_id, manager_id),  
    FOREIGN KEY(job_id) REFERENCES jobs(job_id)  
  
);  
  
CREATE TABLE departments (  
    department_id INT,  
    department_name VARCHAR(20),  
    manager_id INT,  
    location_id DECIMAL(20, 4) UNIQUE,
```

Lec 10-11 (DDL)

```
PRIMARY KEY(department_id, location_id),  
FOREIGN KEY(manager_id) REFERENCES employees(manager_id)
```

```
);
```

```
CREATE TABLE locations (  
  location_id INT PRIMARY KEY,  
  street_address VARCHAR(100),  
  postal_code INT,  
  city VARCHAR(20),  
  state_province VARCHAR(20),  
  country_id INT,
```

```
  FOREIGN KEY(location_id) REFERENCES departments(location_id)
```

```
);
```

```
CREATE TABLE countries (  
  country_id INT PRIMARY KEY,  
  country_name VARCHAR(20),  
  region_id INT UNIQUE
```

```
);
```

```
CREATE TABLE regions (  
  region_id INT PRIMARY KEY,  
  region_name VARCHAR(20),
```

```
  FOREIGN KEY(region_id) REFERENCES countries(region_id)
```

```
);
```

Task B

```
CREATE TABLE Test_1 (  
  ord_num NUMBER(6) NOT NULL,  
  ord_amount NUMBER(12, 2),  
  ord_date DATE NOT NULL,  
  cust_code VARCHAR(6) NOT NULL,  
  agent_code VARCHAR(6) NOT NULL
```

```
);
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>TEST_1</u>	<u>ORD_NUM</u>	NUMBER	-	6	0	-	-	-	-
	<u>ORD_AMOUNT</u>	NUMBER	-	12	2	-	✓	-	-
	<u>ORD_DATE</u>	DATE	7	-	-	-	-	-	-
	<u>CUST_CODE</u>	VARCHAR2	6	-	-	-	-	-	-
	<u>AGENT_CODE</u>	VARCHAR2	6	-	-	-	-	-	-
1 - 5									

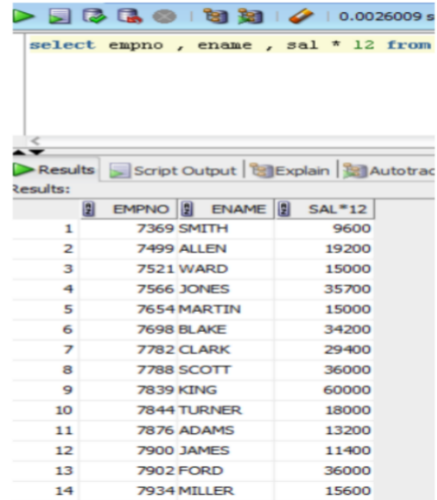
TASK A

- Display the Annual Salary of all the employees.

QUERY:

```
SELECT empno , ename , sal*12
FROM emp ;
```

OUTPUT:



The screenshot shows a SQL query execution window with the query 'select empno , ename , sal * 12 from' entered. Below the query, the 'Results' tab is active, displaying a table with 14 rows of employee data. The columns are EMPNO, ENAME, and SAL * 12.

	EMPNO	ENAME	SAL * 12
1	7369	SMITH	9600
2	7499	ALLEN	19200
3	7521	WARD	15000
4	7566	JONES	35700
5	7654	MARTIN	15000
6	7698	BLAKE	34200
7	7782	CLARK	29400
8	7788	SCOTT	36000
9	7839	KING	60000
10	7844	TURNER	18000
11	7876	ADAMS	13200
12	7900	JAMES	11400
13	7902	FORD	36000
14	7934	MILLER	15600

TASK B

1. Find errors:

Select empno, ename sal x 12 Annual Salary

From emp;

2. Display employee's annual salary with one time bonus of \$100.

3. Display annual compensation as monthly salary plus a monthly bonus of \$100.

4. Display rows in following format:

Monthly Salary

King: 1 month Salary = 5000

5. Display kinds of Jobs available in employee table.

TASK C

- Find all employees whose job is not CLERK and belong to department 20.
- Display the employee's name , job title & salary based on the following criteria:
 - a) If the employee is a salesman, then he should be included in the O/P
 - b) If the employee is a manager, then his salary package must be above 2450.
- Display employee's name , Job titles & salary if the employee is either a salesman or a manager & earns more than 2450.

TASK D

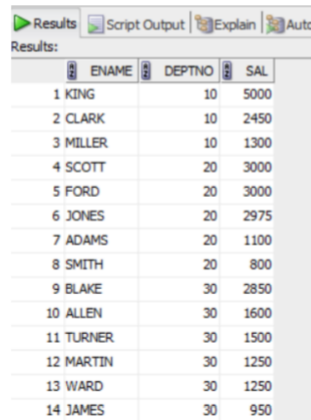
- List the employees having at least two A's in their names.
- List the employees whose names start with S and end at H.
- List the employee whose name has E as the second character.
- Display employ number and job title of all employees who have a job title that contain the string 'MAN' & earn more than 10,000.





TASK E

Display list of employees who are either a clerk or an analyst & who do not earn 1000, 3000,5000.

TASK F

- Display the names of employees according to their seniority.
- Display names and annual salary of all employees, also sort the result based on annual salary in descending order.
- Write a query which produces the following output.



Results:    

	ENAME	DEPTNO	SAL
1	KING	10	5000
2	CLARK	10	2450
3	MILLER	10	1300
4	SCOTT	20	3000
5	FORD	20	3000
6	JONES	20	2975
7	ADAMS	20	1100
8	SMITH	20	800
9	BLAKE	30	2850
10	ALLEN	30	1600
11	TURNER	30	1500
12	MARTIN	30	1250
13	WARD	30	1250
14	JAMES	30	950

Solutions:

Task A

- SELECT empno, ename, sal * 12 FROM emp;

Task B

1. comma missing after ename, x written instead of *, an AS Alias missing before Annual Salary & Annual Salary written without double quotations or an underscore between them.

Correction: SELECT empno, ename, sal * 12 AS annual_salary FROM emp;

2. SELECT ename, (sal*12)+100 FROM emp;
3. SELECT (sal+100)*12 AS "Annual Compensation" FROM emp;
4. SELECT ename || ': 1 month Salary = ' || sal AS "Monthly Salary" FROM emp;
5. SELECT job AS "Kinds of Job" FROM emp;

Task C

- SELECT * FROM emp WHERE job <> 'CLERK' AND deptno = 20;
- - a) SELECT ename, job, sal FROM emp WHERE job = 'SALESMAN';
 - b) SELECT ename, job, sal FROM emp WHERE job = 'MANAGER' AND sal > 2450;
- SELECT ename, job, sal FROM emp WHERE (job = 'SALESMAN' OR job = 'MANAGER') AND sal > 2450;

Task D

- SELECT ename FROM emp WHERE ename LIKE '%A%A%';
- SELECT ename FROM emp WHERE ename LIKE 'S%H';
- SELECT ename FROM emp WHERE ename LIKE '_E%'; [no results]
- SELECT empno, job FROM emp WHERE job LIKE '%MAN%' AND sal > 10000; [no results]

Task E

- SELECT * FROM emp WHERE job IN ('CLERK','ANALYST') AND sal NOT IN (1000, 3000, 5000);

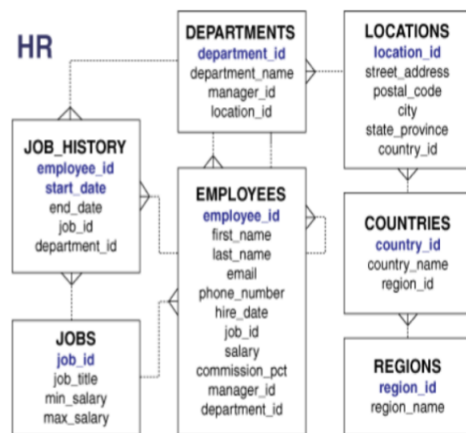
Task F

- SELECT ename FROM emp ORDER BY hiredate;
- SELECT ename, sal*12 AS "Annual Salary" FROM emp ORDER by sal DESC;
- SELECT ename, deptno, sal FROM emp ORDER by deptno, sal DESC;

TASK A

1. Generating new email address

- For each employee, display the first name, last name, and email address. The email address will be composed from the first letter of first name, concatenated with the three first letters of last name, concatenated with *@abc.com*.
- For each employee, display the first name, last name, and email address. The email address will be composed from the first letter of first name, concatenated with the three last letters of last name, concatenated with *@abc.com*.



TASK B

Find the GROSS PAY of all employees using NVL2 function.

```

SELECT ename , NVL2(comm,sal+comm,sal)
FROM emp ;
  
```

	ENAME	NVL2(COMM,SAL+COMM,SAL)
1	SMITH	800
2	ALLEN	1900
3	WARD	1750
4	JONES	2975
5	MARTIN	2650
6	BLAKE	2850
7	CLARK	2450
8	SCOTT	3000
9	KING	5000
10	TURNER	1500
11	ADAMS	1100
12	JAMES	950
13	FORD	3000
14	MILLER	1300

TASK C

1. Display manger id and the salary of the lowest paid employee for that manger, exclude any those whose manger is unknown and sort the result in descending order of the lowest salary.
2. Display the total salary being paid to each job title within each department.
3. Find the total annual salary distributed job wise in the year 81.
4. List the Manager ids & number of employees working for those managers in the ascending order.
5. Find the number of employees who are serving as CLERK?
6. Find the total salary given to the MANAGERS?

TASK D

1. List the departments where at least two employees are working.
2. List the number of employees in each department where the number of employees exceeds 3.
3. Find out the least 5 earners of the emp table.

Solutions:

Task A

- SELECT first_name, last_name, SUBSTR(LOWER(first_name), 1, 1) || SUBSTR(LOWER(last_name), 1, 3) || '@abc.com' AS Email_Address FROM employees;
- SELECT first_name, last_name, SUBSTR(LOWER(first_name), 1, 1) || SUBSTR(LOWER(last_name), -3, 3) || '@abc.com' AS Email_Address FROM employees;

NOTE: [The above two queries of Task A will only work in HR Account of Oracle]

Task B

SELECT ename, NVL2(comm, comm+sal, sal) AS gross_pay_of_all_employees FROM emp;

Task C

1. SELECT mgr, min(sal) AS lowest_paid_employee FROM emp WHERE mgr is NOT NULL GROUP BY mgr ORDER BY min(sal) DESC;
2. SELECT deptno, job, SUM(sal) AS total_salary FROM emp GROUP BY deptno, job ORDER BY deptno;
3. SELECT job, sum(sal*12) FROM emp WHERE TO_CHAR(hiredate, 'YYYY') = '1981' GROUP BY job;
4. SELECT mgr, COUNT(empno) AS no_of_employees FROM emp WHERE mgr IS NOT NULL GROUP BY mgr ORDER BY mgr ASC;
5. SELECT count(empno) FROM emp WHERE job = 'CLERK';
6. SELECT SUM(sal) FROM emp WHERE job = 'MANAGER';

Task D

1. SELECT deptno, count(job) FROM emp GROUP BY deptno, job HAVING count(job) >= 2;
2. SELECT deptno, count(empno) FROM emp GROUP BY deptno HAVING count(empno) > 3;
3. SELECT * FROM emp WHERE ROWNUM < 6 ORDER by sal ASC;