DDL Practice

1. Write necessary **MySQL** DDL commands that will generate the following database:

Tablename: table1		
Column Name	**Datatype**	**Other Constraints**
colA	int	AUTO_INCREMENT PRIMARY KEY
colB	varchar(40)	NULL
colC	datetime	Default "2020-01-01 00:00:00"
foreign_colA - refers to the **colA** column of this table	int	NULL
foreign_colP - refers to the **colP** column of table table2	varchar(30)	NULL

Tablename: table2		
Column	**Datatype**	**Other Constraints**
colP	varchar(30)	PRIMARY KEY
colQ	char(10)	UNIQUE
colR	double	NOT NULL
foreign_colA - refers to the **colA** column of table1	int	NULL

Important points to consider:

Here, **table1** has foreign key from **table2** and also **table2** has foreign key from **table1**.
So which table you will create first?

As you know first we need to create the parent table containing the Primary Key, then we can build the 2nd table(child table) so that this 2nd table can refer to the 1st table's PK column. But in this case both table depends on each other,
So what will you do?

>> First create any one of the tables (suppose, **table1**) and during the creation of **table1** skip the foreign key constraint section cause **table2** doesn't exist at this time. So, this **table1** is incomplete.

>> Then create the **table2**, this time no problem will occur as **table1** already exists.

>> Finally, alter the **table1** to add the skipped FK constraint. Now **table2** exists, so this constraint will create no problem.

2. Insert some sample data to each of the tables.
3. Update the existing data of table1 or, table2.
4. Delete some data from table1 or, table2.

HR Schema - ERD
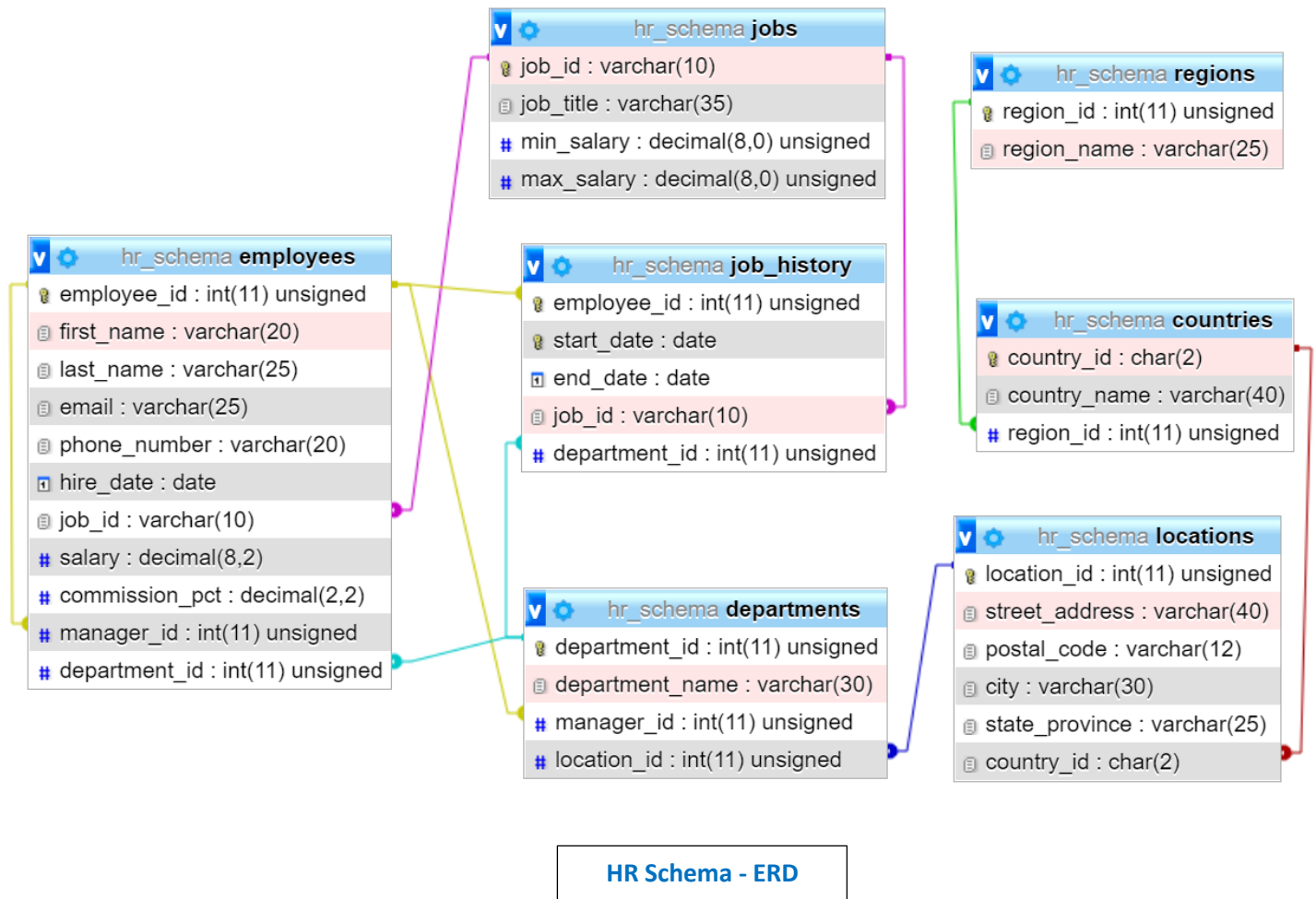
Basic Search Operations (SELECT, WHERE, ORDER BY, LIMIT clauses)

```
SELECT [DISTINCT] col1, col2*5 [AS 'newcol2'], col3+col4 [AS 'newcol3'],
                                    function(col5) [AS 'newcol4'], ... ... ...
FROM tablename
[WHERE condition]
[ORDER BY col1 [ASC|DESC], col2 [ASC|DESC], ... ...]
[LIMIT [offset,] rowcount]
```

1. Show all data from **countries** table.
2. Show all data from **employees** table.
3. Show all data from **departments** table.
4. Show all data from **job_history** table.

LIKE / NOT LIKE

1. Show those employee details whose first name starts with the letter **s.**
2. Show those employee details whose first name doesn't start with the letter **s.**
3. Show those employee details whose first name ends with the letter **a.**
4. Show those employee details whose first name contains **da** as substring.
5. Show those employee details whose first name starts with **s** and ends with **a.**
6. Show those employee details whose first name either starts with **s** or starts with **m.**
7. Show those employee details whose first name contains the letter **o** and **a.**

---

8. Show those employee details whose first name contains the letter **o** followed by the letter **a.**
9. Show those employee details whose first name consists of exactly 3 characters.
10. Show those employee details whose first name consists of minimum 3 characters.
11. Show those employee details whose first name contains the letter **a** from the second last position.

IN() / NOT IN( ) / OR, ||

1. Show those country details whose country_id is **AU/BR/CN/JP**.
2. Show those department details whose manager_id is not **204/100/145**.
3. Show those employee details whose job_id is **ST_MAN/IT_PROG**.
4. Show those employee details who does not work in the department_id **100/30/90**.
5. Show those location details where the postal_code is either **2901/50090**.
6. Show those location details where the city name is either **Roma/Venice/Tokyo**.

BETWEEN … AND… / NOT BETWEEN … AND … / AND, &&

1. Show those department details whose location_id is within the range 1000 to 2000 inclusive.
2. Show those employee details whose salary is within the range 10000 to 20000 inclusive.
3. Show those employee details whose hire_date is within the range '1987-01-01' to '1987-06-30' inclusive.
4. Show those employee details whose department_id is not within the range 50 to 60 inclusive.
5. Show those job details where the difference between max_salary and min _salary is within the range 5000 to 10000 inclusive.
6. Show those job_history details where the end_date is within the range '1998-12-01' to '1998-12-31' inclusive.

CASE WHEN … WHEN … ELSE … END statement

1. Show all the past employees employee id, start date, job id and his group name from the job_history table:
   Determine the group name as below:

Start Date	Group Name
on or before 1989-12-31	C
between 1994-12-31 to 1900-01-01 inclusive	B
after 1995-01-01 inclusive	A

2. Show the department id, department name, location id and location group name from the departments table:
   Determine the location group name as below:

Location Id	Location Group Name
less than 1200	C
between 2000 to 1200 inclusive	B
greater than 2000	A

3. From the jobs table show the job id, job title, and job group name.
   Determine the job group name based on the following criteria:

JOB_TITLE	Job Group Name
job title containing the word "president"	President
job title containing the word "manager"	Manager
all other job titles	Other

4. From the locations table show the location id, city name, state province, and location group name.

Mohammad Imam Hossain, Email: imambuet11@gmail.com

Group the locations based on the following criteria:

City Name	Location Group Name
Tokyo, Venice, Toronto, Oxford, Singapore	A
Mexico City, London, Sydney, Seattle, Beijing	B
all other cities	C

5. Show the EMPLOYEE_ID, START_DATE, END_DATE and SENIORITY_LEVEL from the job_history table where the SENIOR_LEVEL will be calculated as follow:

Difference between the END_DATE and START_DATE	SENIORITY_LEVEL
diff > 1200 days	A
600 days < diff < 1200 days	B
diff < 600 days	C

6. Show the JOB_ID, JOB_TITLE and SALARY_RANGE_CLASS from the jobs table where the SALARY_RANGE_CLASS will be calculated as follow:

Difference between MAX_SALARY and MIN_SALARY values	SALARY_RANGE_CLASS
diff > 20k	A
10K < diff <20k	B
diff < 10K	C

Numerical and String Functions

1. Show all the employees employee id and their short name in lowercase format.
   Short name format: first 3 letters from the first name followed by an underscore and then followed by the first 3 letters of the last name.
2. Show all those employee details whose name is a palindrome.
3. Show all the employees employee id and email (i.e. add '@gmail.com' at the end of each email).
4. Show all the employees first name and phone number.
   Phone number format: 515.xxx.xxx7 i.e. only show the first 4 characters and the last character and hide all the intermediate characters with xxx.xxx
5. Show all the employees employee id, email and full name.
   Full name format: first_name<SPACE>last_name
   Also show the full name in 20 characters if necessary right pad with necessary no of spaces.
6. Show those location details from locations table whose postal code consists at most 5 characters and the first two digits of its postal code is between 50 to 99 inclusive.
7. Show all the employees employee id, first name and his salary in "10 thousand 5 hundred and 12 taka only" format.
8. For each job, show the job id, job title and how much greater the max_salary from its min_salary in percentage format.
   Note: Show the output in 2 decimal points
   %greater=(max_salary-min_salary)*100/min_salary
9. Show all those job details from jobs table whose salary range (i.e. max_salary-min_salary) is greater than 8000 and the job title contains the word 'Manager'.
10. Show all the employees employee id, and his yearly total gross salary.
    Note: Show the floor value of the total salary
    yearly total salary = salary * 12* (1+(commission_pct/100))]

11. Show those department details from departments table whose tens digits of location id is within the range 5 to 9 inclusive.

Date and Time Functions

1. Show all the employees' email, hire date in "January 4th, 1987" format.
2. Show all the employees' email, hire date in "Jan 1987, 04" format.
3. Show all the employees' email, hire date in "1st Aug, 87 05:10 PM" format.
4. Show all the employees' email, hire date in "15 Jan, 2019 Tuesday 14:10" format.

5. Show those employees first name, email, phone number who is hired after the date "05 May, 1987 00:00 AM".
6. Show those employees first name, email, phone number who is hired before the date "1st June 1987 11:01 PM".

7. Show all the employees employee id and his current job experience (till today) in number of years format.
   Note: Show the no of years in 3 decimal points.
8. Show all the employees employee id, email and his current job experience in "10 years, 06 months and 15 days" format.
   Note: show the months in two digits format if necessary left pad with 0.
9. Show all those employees employee id from the job_history table whose job experience is greater than 5 years.

10. For each job_history, show how many days an employee has served during his last month of retirement.
11. For each employee, show how many days an employee has served during his first month of joining.

12. Show all those employee details who have been hired on the leap day(29th Feb) of any leap year.
13. Show all those employee details whose hiring month is either 2/4/6/8.
14. Show all those employee details who have joined either in the year of 1997 or in the month of February (of any year).

15. Show all the employees first name, department id, manager id and his updated join date that is one week before the real join date.
16. From the employees table, show all the employees employee id, join date and estimated retirement date that is 35 years after his join date.

ORDER BY clause

1. Show all the employees first name, last name, email, hire date, salary in descending order of salary. If multiple employees receive the same salary then also sort them based on the alphabetical order of their first name.
2. Show all the employees employee id and their join date in such a way that the senior most employee comes first. If multiple employees have the same join date then also sort them based on the descending order of their department id.
3. Show all the employees first name, email and phone number. Order the output based on the descending order of first 3 digits of their phone number.
4. Show all the employees employee id, email, hire year (only the year portion) and hire month (show the full month name). Show the output from most recent hired employee to old employees.
5. Show all the job_history details in such a way that senior most employee data comes first and if multiple employees have the same start date then also sort them based on the descending order of their end date.

6. Show all the jobs from jobs table where the highest salary range (i.e. max_salary-min_salary) job data comes first.

DISTINCT clause

1. Show all the distinct manager_ids from employees table.
2. Show all the distinct job_ids from the employees table.
3. Show all the distinct country_ids from the locations table.
4. Show all the distinct job_ids and department_ids from employees table.

LIMIT clause

1. Show the highest salary holder employee details from the employees table.
2. Show the top 10 most experienced employee details from the employees table.
3. Show the 2nd lowest salary range (i.e. max_salary-min_salary) job details from the jobs table.
4. Show the top 3 lowest salary holder employee details from department number 60.
5. Among the employees supervised by manager id 108, find out the 2nd highest salary holder employee details.
6. Among the employees whose job type is 'ST_CLERK', show the highest experienced employee id from the job_history table.

Aggregate Operations (GROUP BY, HAVING clauses)

```
SELECT col1, col2, groupfn(col3), groupfn1(col4), ... ...
FROM tablename
[WHERE condition]
GROUP BY col1, col2
HAVING condition
[ORDER BY col1 [ASC|DESC], col2 [ASC|DESC], ... ...]
[LIMIT [offset,] rowcount]
```

1. Show the total no of employees, their total salary, their average salary, their maximum salary, their minimum salary from employees table.
2. Show the maximum and minimum experienced employees hire dates from employees table.
3. Show the maximum experienced employee hire date working in department number 50 from employees table.
4. Show the number of departments located in location id 1700 from departments table.
5. Show the most recent retired employee's end date working in department number 80 from job history table.
6. Show the maximum and minimum value of min_salary column, maximum and minimum value of max_salary column from jobs table.
7. Count the number of employees managed by manager id 114 from employees table.
8. Count the total number of distinct job_ids from employees table.
9. Count the distinct number of countries from locations table.
10. Count the total number of locations located in 'US' from locations table.
11. Show the maximum and minimum salary range value (i.e. salary range = max_salary - min_salary) from jobs table.
12. Count the number of employees whose employee id is greater than his manager id.

1. Show each region_id and corresponding no of countries in that region from countries table.
2. Show the location_id and corresponding no of departments in that location from departments table.
3. For each department_id, show the no of employees in that department from employees table.
4. For each manager_id, show the no of employees under his supervision from employees table.
5. For each job_id, show the no of employees in that job type from employees table.
6. For each department_id, show the no of managers from that department using employees table.
7. Count the total number of employees joined in the even month and total number of employees joined in the odd number months from the employees table.

8. Show the department wise total no of employees, maximum and minimum salary in that department, average and total salary provided by that department from the employees table.
9. For each year, show the total no of employees who were hired during that year from the employees table.
10. Show the total no of jobs within 0k to 10k, 10k to 20k and so on salary ranges(max_salary-min_salary) groups from the jobs table.
11. For each country_id, show the total no of locations in that country from the locations table.
12. For each city, show the total no of locations in that city from the locations table.
13. Group and count employees based on the first letter of their names. (max 26 groups as 26 alphabets)

14. For each job_id and each department, show the total no of employees in that group from the employees table.
15. For each year and each month, show the total no of employees who have left their jobs from the job_history table.

---

1. Show only those manager_ids who handle more than 5 employees.
2. Show only those department_ids where in total salary expense is more than 100000 dollar.
3. Count the total no of employees for each department. Don't consider employees of job_id "AD_PRESS" and also consider only those departments where total no of employees is greater than 5.
4. Group employees based on the first 3 digit of their phone number. Avoid employees from department no 10/20/60 and also avoid those groups where total salaries of employees is less than 50000.

5. For each year and each month, count total number of employees joined from employees table. Don't consider those year and months where total number of hired employees are less than 20.
6. For each country and each city, count total number of locations from locations table. Don't consider locations from city 'US' and also don't consider those country and city having less than 5 locations.

Table Join Operations (JOIN, LEFT JOIN clauses)

```
SELECT t1.*, t2.*, t3.col1, t3.col2, ... ...
FROM tablename1 AS t1

    JOIN
    tablename2 AS t2
    ON join_condition

    JOIN
    tablename3 AS t3
    ON join_condition

[WHERE condition]
...
...
```

1. Show the region_name and corresponding country_name
2. Show the department_name and corresponding country_name.
3. Show the employee_name and his job place country_name.
4. Show the employee_name and his job_title.

5. Show the employee_name and his manager_name
6. Show the department_name and the manager_name of corresponding department.
7. Show the employee_id, his salary, his manager_id, his manager_name, his manager_salary.
8. Show the employee_id, his join_date, his manager_id, his manager_name, his manager_salary.
9. Show the manger_name and his manager_name (manager of manager).
10. Show the employee name and his manager name only for those employees who have joined after this manager.
11. Show the employees name and other employees name who receives higher salary than him

---

Mohammad Imam Hossain, Email: imambuet11@gmail.com

12. Show the employees name and other employees name who is hired after him.

13. For each region, show the region_name and total no of employees in that region.
14. For each job, show the job_title and total no of employees.
15. For each country, show the total no of departments in that country.
16. For each department, show the department_name and corresponding total no of ex-employees (job_history table) from that department.

17. For each manager, show the manager_name and total no of employees under his supervision.
18. For each manager, show the manager_name and total no of employees under his supervision who receives higher salary than him.
19. Show the employee name and no of employees who receives lower salary than him.
20. Show the employee name and no of employees who is hired before him.

Mohammad Imam Hossain, Email: imambuet11@gmail.com