HONG KONG INSTITUTE OF VOCATIONAL EDUCATION

## Laboratory 2: Data Retrieval and Manipulation

**Module Intended Learning Outcome (#3):**

On completion of the module, students are expected to be able to:

* Develop database queries to perform typical data definition and data manipulation operations in relational database systems.

## Lesson Intended Learning Outcome:

On completion of this lab, students are expected to be able to:

* use Navicat's Query Builder to create SELECT queries.
* Add WHERE condition to fetch specific data from a database.
* Apply aggregate functions and grouping records.

**Theme Park database**

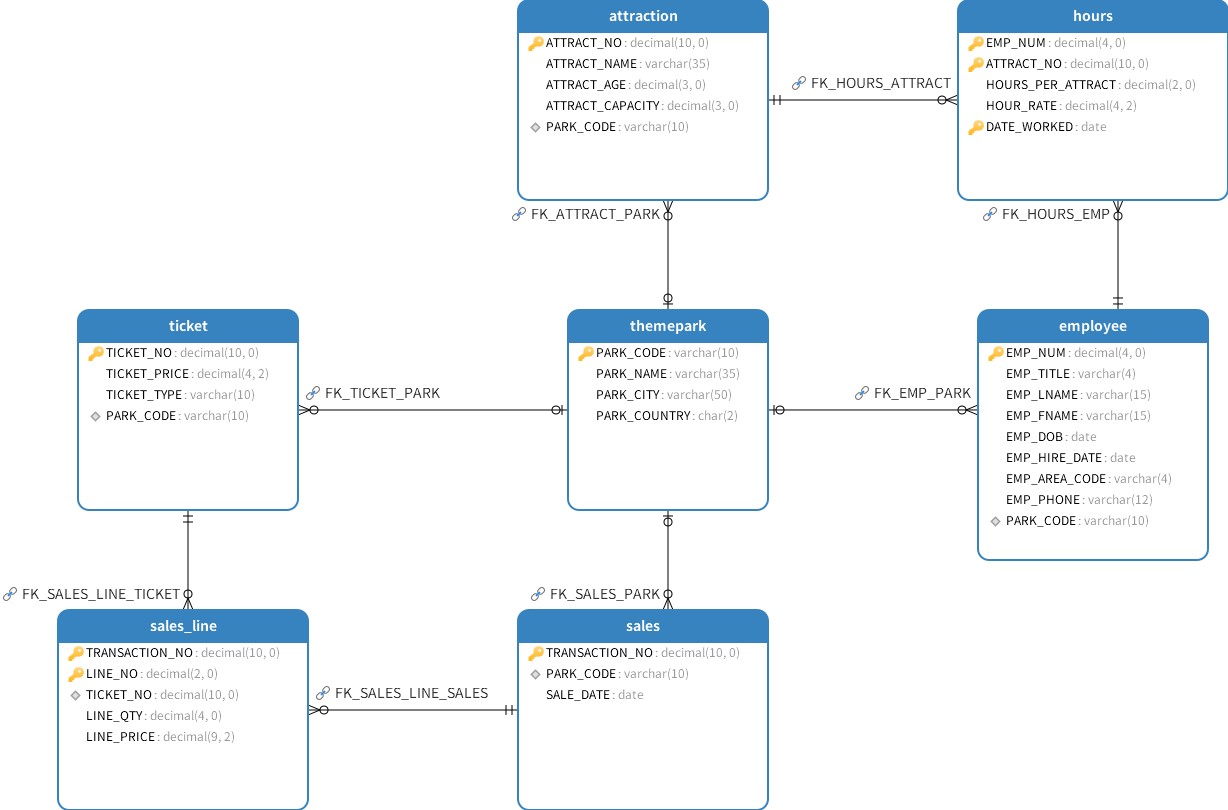
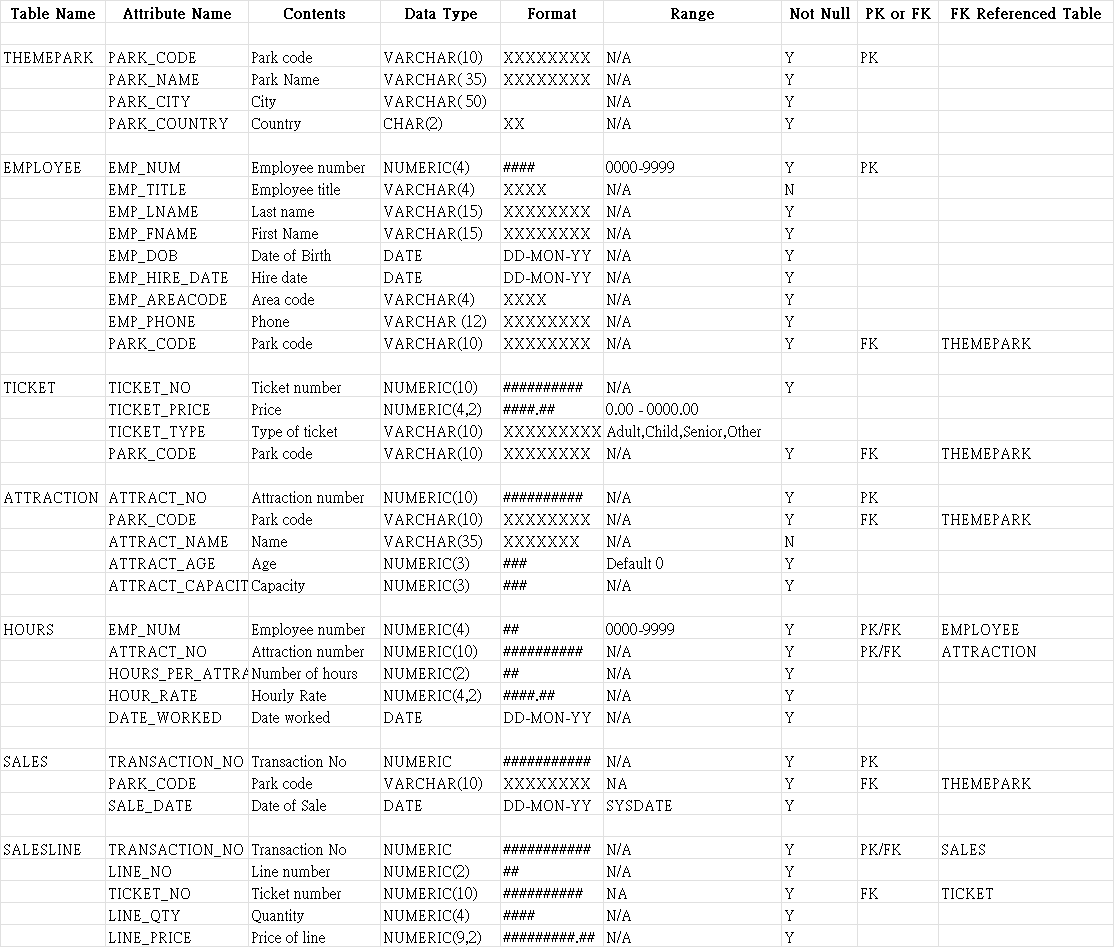
****

Figure 1 Entity Relationship Diagram



Data Dictionary of Theme Park database

# TASK:

1. Connect to MySQL database using Navicat.
2. Use THEME\_PARK database.
3. Create a query to display all employee first name, last name and birthday.
4. Create a query to display all employees who were hired before 1995.
5. Create a query to display all Theme Parks except those in the UK.
6. Create a query to display all the sales that occurred on the 18th May 2007.
7. Create a query to display the ticket prices between €20 AND €30.
8. Display all attractions that have a capacity of more than 60 at the Theme Park FR1001.
9. Write a query to display the hourly rate for each attraction where an employee had worked, along with the hourly rate increased by 20%. Your query should only display the ATTRACT\_NO, HOUR\_RATE and the HOUR\_RATE with the 20% increase.
10. Write a query to calculate the number of unique park code that exist in the TICKET table, replace calculated column with ‘Park\_Count’.

**Sorting and Group by clause**

1. Display the employee numbers of all employees and the total payment (Hour rate multiply Hour per attract) they have worked. Use alias ‘Total payment’ for calculated column. Sort the output according employee number descending.
2. Write a query to show the transaction numbers and AVERAGE line prices (use the SALES\_LINE table). Display average value that are greater than €50 only.
3. Display the employee numbers of all employees and the total hours they have worked. Use ‘Total Hours’ for calculated column. Sort the output according employee number descending.
4. Show the attraction number and the minimum and maximum hourly rate for each attraction. Use alias MIN and MAX for calculated column.
5. Display Transaction Number, Sale date information from the SALES table. Calculate the number of sales after 1 January 2007. Sort the result in descending order of the sale date.
6. Using the TICKET table, write a query to display the park code and the average ticket price. Limiting the average ticket price greater or equal to 20. Replace calculated column name using ‘AVERAGE PRICE’.

**Joining Tables**

1. Display the attractions that have no employees working on them.
2. Display the employee names along with the theme park names where they work.
3. Use the cross join to display all rows in the EMPLOYEE and HOURS tables. How many rows were returned?
4. Write a query to display the attraction number, employee first and last names and the date they worked on the attraction. Order the results by the date worked.
5. Display the park names and total sales for Theme Parks who are located in the country ‘UK’ or ‘FR’.
6. Write a query to display the names of attractions that currently have not had any employees working on them.
7. List the sale date, line quantity and line price of all transactions on the 18th May 2007. (Hint: Remember the format of MySQL dates is ‘2007-05-18’)

END