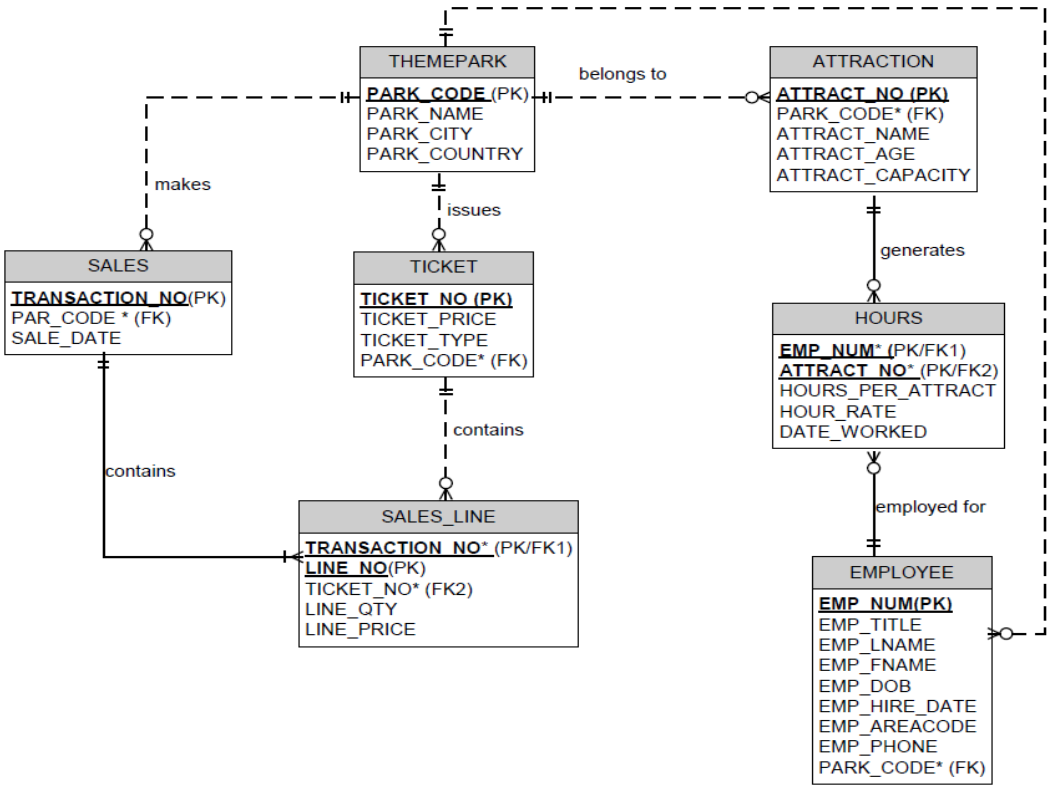
Lab 7: Retrieving Data from Multiple Tables (Use Themepark db)

Introduction to Joins

The relational join operation merges rows from two or more tables and returns the rows with one of the following conditions:

- Have common values in common columns (natural join)
- Meet a given join condition (equality or inequality)
- Have common values in common columns or have no matching values (outer join)



ERD

For example,

suppose you want to join the two tables THEMEPARK and TICKET. Because PARK_CODE is the foreign key in the TICKET table and the primary key in the THEMEPARK table, the link is established on PARK_CODE.

It is important to note that when the same attribute name appears in more than one of the joined tables, the source table of the attributes listed in the SELECT command sequence must be defined. To join the THEMEPARK and TICKET tables, you would use the following, which produces the output shown.

```
SELECT THEMEPARK.PARK_CODE, PARK_NAME, TICKET_NO, TICKET_TYPE,
                    TICKET_PRICE
FROM THEMEPARK, TICKET
```

WHERE THEMEPARK.PARK_CODE = TICKET.PARK_CODE;

```
SELECT THEMEPARK.PARK_CODE, PARK_NAME, TICKET_NO, TICKET_TYPE, TICKET_PRICE
FROM THEMEPARK, TICKET
WHERE THEMEPARK.PARK_CODE = TICKET.PARK_CODE;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PARK_CODE	PARK_NAME	TICKET_NO	TICKET_TYPE	TICKET_PRICE
FR1001	FairyLand	13001	Child	18.99
FR1001	FairyLand	13002	Adult	34.99
FR1001	FairyLand	13003	Senior	20.99
UK3452	PleasureLand	88567	Child	22.50
UK3452	PleasureLand	88568	Adult	42.10
UK3452	PleasureLand	89720	Senior	10.99

As you examine the preceding query, note the following points:

- The FROM clause indicates which tables are to be joined. If three or more tables are included, the join operation takes place two tables at a time, starting from left to right. For example, if you are joining tables T1, T2, and T3, first table T1 is joined to T2; the results of that join are then joined to table T3.
- The join condition in the WHERE clause tells the SELECT statement which rows will be returned. In this case, the SELECT statement returns all rows for which the PARK_CODE values in the PRODUCT and VENDOR tables are equal.
- The number of join conditions is always equal to the number of tables being joined minus one.

  Table number – 1 = bilangan join predicate.

  For example, if you join three tables (T1, T2, and T3), you will have two join conditions (j1 and j2). All join conditions are connected through an AND logical operator. The first join condition (j1) defines the join criteria for T1 and T2. The second join condition (j2) defines the join criteria for the output of the first join and table T3.

- Generally, the join condition will be an equality comparison of the primary key in one table and the related foreign key in the second table.

1. Execute the following query and check your results with those shown. Then modify the SELECT statement and change THEMEPARK.PARK_CODE to just PARK_CODE. What happens?

   SELECT THEMEPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME,
   ATTRACT_CAPACITY

   FROM THEMEPARK, ATTRACTION

   WHERE THEMEPARK.PARK_CODE = ATTRACTION.PARK_CODE; → join predicate

```
SELECT THEMEPARK.PARK_CODE, PARK_NAME, ATTRACT_NAME, ATTRACT_CAPACITY
FROM THEMEPARK, ATTRACTION
WHERE THEMEPARK.PARK_CODE = ATTRACTION.PARK_CODE;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

PARK_CODE	PARK_NAME	ATTRACT_NAME	ATTRACT_CAPACITY
FR1001	FairyLand	ThunderCoaster	34
FR1001	FairyLand	SpinningTeacups	62
FR1001	FairyLand	FlightToStars	24
FR1001	FairyLand	Ant-Trap	30
ZA1342	GoldTown	NULL	40

Joining tables with an alias

An alias may be used to identify the source table from which the data are taken. For example, the aliases P and T can be used to label the THEMEPARK and TICKET tables.

Example :

SELECT P.PARK_CODE, PARK_NAME, TICKET_NO, TICKET_TYPE, TICKET_PRICE

FROM THEMEPARK P, TICKET T
WHERE P.PARK_CODE =T.PARK_CODE; → join predicate

2. Write a query that displays the employees first and last name (EMP_FNAME and EMP_LNAME), the attraction number (ATTRACT_NO) and the date worked. **Hint**: You will have to join the HOURS and the EMPLOYEE tables.

```
SELECT EMP_FNAME,EMP_LNAME, H. ATTRACT_NO
FROM EMPLOYEE E, HOURS H
WHERE E.EMP_NUM = H.EMP_NUM;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EMP_FNAME	EMP_LNAME	ATTRACT_NO
Emma	Calderdale	10034

Join USING

A second way to express a join is through the USING keyword.

That query returns only the rows with matching values in the column indicated in the USING clause—and that column must exist in both tables. The syntax is:

SELECT *column-list* FROM *table1* JOIN *table2* USING (*common-column*)

3. To see the JOIN USING query in action, let's perform a join of the SALES and SALEs_LINE tables by writing:

SELECT TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY, LINE_PRICE
FROM SALES JOIN SALES_LINE USING (TRANSACTION_NO);

```
SELECT TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY, LINE_PRICE
FROM SALES JOIN SALES_LINE USING (TRANSACTION_NO);
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TRANSACTION_NO	SALE_DATE	LINE_NO	LINE_QTY	LINE_PRICE
12781	2007-05-18	1	2	69.98
12781	2007-05-18	2	1	14.99
12782	2007-05-18	1	2	69.98

4. Rewrite the query you wrote in task **2** so that the attraction name (ATTRACT_NAME located in the ATTRACTION table) is also displayed. Express the joins through the USING keyword. Hint: You will need to join three tables.

SELECT EMP_FNAME, EMP_LNAME, H. ATTRACT_NO, A.ATTRACT_NAME
FROM EMPLOYEE E, HOURS H, ATTRACTION A
WHERE E.EMP_NUM = H.EMP_NUM
AND A.ATTRACT_NO = H.ATTRACT_NO;

```
SELECT EMP_FNAME, EMP_LNAME, H. ATTRACT_NO, A.ATTRACT_NAME
FROM EMPLOYEE E, HOURS H, ATTRACTION A
WHERE E.EMP_NUM = H.EMP_NUM
AND  A.ATTRACT_NO = H.ATTRACT_NO;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

EMP_FNAME	EMP_LNAME	ATTRACT_NO	ATTRACT_NAME
Emma	Calderdale	10034	ThunderCoaster

Join ON

The previous two join styles used common attribute names in the joining tables. Another way to express a join when the tables have no common attribute names is to use the JOIN ON operand.

SELECT *column-list* FROM *table1* JOIN *table2* ON *join-condition*

The following example performs a join of the SALES and SALES_LINE tables, using the ON clause.

SELECT SALES.TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY, LINE_PRICE
FROM SALES JOIN SALES_LINE ON SALES.TRANSACTION_NO =
SALES_LINE.TRANSACTION_NO;

```
363 •  SELECT SALES.TRANSACTION_NO, SALE_DATE, LINE_NO, LINE_QTY, LINE_PRICE
364    FROM SALES JOIN SALES_LINE ON SALES.TRANSACTION_NO = SALES_LINE.TRANSACTION_NO;
365
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TRANSACTION_NO	SALE_DATE	LINE_NO	LINE_QTY	LINE_PRICE
12781	2007-05-18	1	2	69.98
12781	2007-05-18	2	1	14.99
12782	2007-05-18	1	2	69.98

SUMMARY:

There are many ways to write query for retrieving data / joining multiple tables.

You have learn three ways- WHERE, JOIN-ON, JOIN-USING. You also can use alias to simplify the qualifier.

Now lets look at the following example for all three different types of joining query which produces same output.

JOIN TABLE	WHERE	JOIN-ON	JOIN-USING
2 tables	SELECT P.PARK_CODE, PARK_NAME,TICKET_PRICE FROM PARK P, TICKET T WHERE P.PARK_CODE = T.PARK_CODE;	SELECT P.PARK_CODE, PARK_NAME,TICKET_PRICE FROM PARK P JOIN TICKET T ON P.PARK_CODE = T.PARK_CODE;	SELECT P.PARK_CODE, PARK_NAME,TICKET_PRICE FROM PARK P JOIN TICKET T USING (PARK_CODE);
3 tables	SELECT EMP_FNAME, H.ATTRACT_NO, A.ATTRACT_NAME FROM EMPLOYEE E, HOURS H, ATTRACTION A WHERE E.EMP_NUM = H.EMP_NUM AND A.ATTRACT_NO = H.ATTRACT_NO;	SELECT EMP_FNAME, H.ATTRACT_NO, A.ATTRACT_NAME FROM EMPLOYEE E JOIN HOURS H ON E.EMP_NUM = H.EMP_NUM JOIN ATTRACTION A ON A.ATTRACT_NO = H.ATTRACT_NO;	SELECT EMP_FNAME, H.ATTRACT_NO, A.ATTRACT_NAME FROM EMPLOYEE E JOIN HOURS H USING (EMP_NUM) JOIN ATTRACTION A USING (ATTRACT_NO);

EXERCISE:

1. 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.

2. Display the park names and total sales for Theme Parks who are located in the country ‘UK’ or ‘FR’.

3. Write a query to display the names of attractions that hour rate more than 6.

4. Produce a report that lists employee number, employees' last names, first names, and date worked. List data for only park code between FR1001 and UK3452, and exclude employee number 102 from the list. Sequence the report on first name within last name, within date worked, within employee number.
(Refer note simple query for cascade order)

5. For all projects that have a park code beginning with UK, list park code, park name, and attraction name. List identical rows once. Order the list by park code and then by attraction name.

6. Which employees are assigned to attract number 10034? List employee number, last name, and attract number. Order the list by employee

7. List employee number, employee date of Birth, employee age, and the park name he/she handle. Sort the list based on the employee age.