

DATABASE MANAGEMENT SYSTEMS
2016-2017 FALL SEMESTER
LABORATORY MANUAL

Experiment 6

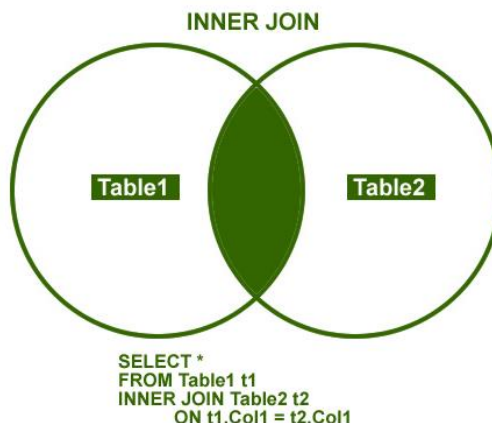
JOINS

SQL Server (Transact-SQL) JOINS are used to retrieve data from multiple tables. A SQL Server JOIN is performed whenever two or more tables are joined in a SQL statement. There are 4 different types of SQL Server joins:

- SQL Server INNER JOIN (or sometimes called simple join)
- SQL Server LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- SQL Server RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)
- SQL Server FULL OUTER JOIN (or sometimes called FULL JOIN)

1. INNER JOIN (simple join)

SQL Server INNER JOINS return all rows from multiple tables where the join condition is met. It is the most common type of join. This join returns rows when there is at least one match in both the tables.



Example

We have a table called suppliers with two fields (supplier_id and supplier_name). And, we have another table called orders with three fields (order_id, supplier_id, and order_date).

supplier_id	supplier_name
10000	IBM
10001	Hewlett Packard
10002	Microsoft
10003	NVIDIA

order_id	supplier_id	order_date
500125	10000	2003/05/12
500126	10001	2003/05/13
500127	10004	2003/05/14

If we run the SQL Server SELECT statement (that contains an INNER JOIN) below:

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date
```

```
FROM suppliers
INNER JOIN orders
ON suppliers.supplier_id = orders.supplier_id;
```

OR

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date
FROM suppliers, orders
WHERE suppliers.supplier_id = orders.supplier_id;
```

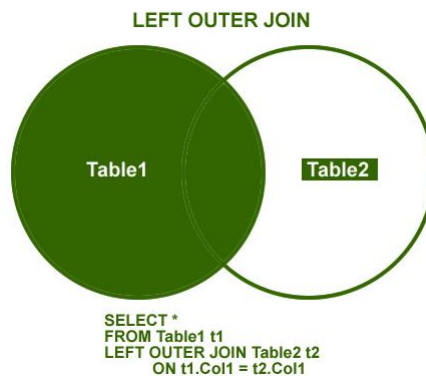
This SQL Server INNER JOIN example would return all rows from the suppliers and orders tables where there is a matching supplier_id value in both the suppliers and orders tables.

supplier_id	name	order_date
10000	IBM	2003/05/12
10001	Hewlett Packard	2003/05/13

The rows for Microsoft and NVIDIA from the supplier table would be omitted, since the supplier_id's 10002 and 10003 do not exist in both tables. The row for 500127 (order_id) from the orders table would be omitted, since the supplier_id 10004 does not exist in the suppliers table.

2. **LEFT OUTER JOIN**

LEFT OUTER JOIN returns all rows from the LEFT-hand table specified in the ON condition and only those rows from the other table where the joined fields are equal (join condition is met).



In some databases, the LEFT OUTER JOIN keywords are replaced with LEFT JOIN.

Example

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date
FROM suppliers
LEFT OUTER JOIN orders
ON suppliers.supplier_id = orders.supplier_id;
```

This LEFT OUTER JOIN example would return all rows from the suppliers table and only those rows from the orders table where the joined fields are equal.

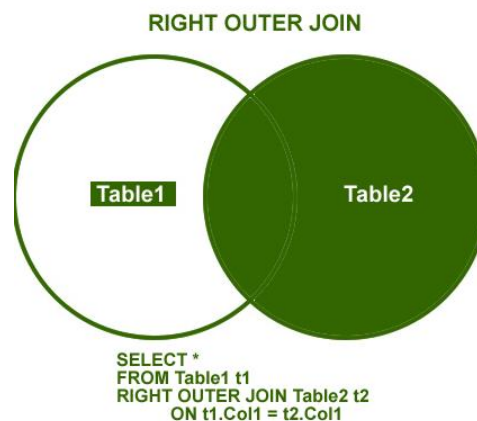
If a supplier_id value in the suppliers table does not exist in the orders table, all fields in the orders table will display as <null> in the result set.

supplier_id	supplier_name	order_date
10000	IBM	2003/05/12
10001	Hewlett Packard	2003/05/13
10002	Microsoft	<null>
10003	NVIDIA	<null>

The rows for *Microsoft* and *NVIDIA* would be included because a LEFT OUTER JOIN was used. However, you will notice that the order_date field for those records contains a <null> value.

3. **RIGHT OUTER JOIN**

RIGHT OUTER JOIN returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where the joined fields are equal (join condition is met).



The SQL Server RIGHT OUTER JOIN would return the all records from Table2 and only those records from Table1 that intersect with Table2.

Example

```
SELECT orders.order_id, orders.order_date, suppliers.supplier_name
FROM suppliers
RIGHT OUTER JOIN orders
ON suppliers.supplier_id = orders.supplier_id;
```

This RIGHT OUTER JOIN example would return all rows from the orders table and only those rows from the suppliers table where the joined fields are equal.

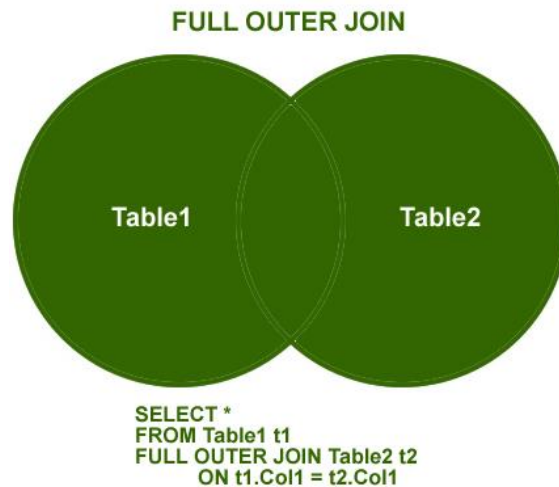
If a supplier_id value in the orders table does not exist in the suppliers table, all fields in the suppliers table will display as <null> in the result set.

order_id	order_date	supplier_name
500125	2013/08/12	Apple
500126	2013/08/13	Google
500127	2013/08/14	<null>

The row for 500127 (order_id) would be included because a RIGHT OUTER JOIN was used. However, you will notice that the supplier_name field for that record contains a <null> value.

4. **FULL OUTER JOIN**

FULL OUTER JOIN returns all rows from the LEFT-hand table and RIGHT-hand table with nulls in place where the join condition is not met.



Example

```
SELECT suppliers.supplier_id, suppliers.supplier_name, orders.order_date  
FROM suppliers  
FULL OUTER JOIN orders  
ON suppliers.supplier_id = orders.supplier_id;
```

This **FULL OUTER JOIN** example would return all rows from the suppliers table and all rows from the orders table and whenever the join condition is not met, <null> would be extended to those fields in the result set.

If a `supplier_id` value in the suppliers table does not exist in the orders table, all fields in the orders table will display as <null> in the result set. If a `supplier_id` value in the orders table does not exist in the suppliers table, all fields in the suppliers table will display as <null> in the result set.

supplier_id	supplier_name	order_date
10000	IBM	2013/08/12
10001	Hewlett Packard	2013/08/13
10002	Microsoft	<null>
10003	NVIDIA	<null>
<null>	<null>	2013/08/14

The rows for Microsoft and NVIDIA would be included because a **FULL OUTER JOIN** was used. However, you will notice that the `order_date` field for those records contains a <null> value.

The row for `supplier_id` 10004 would be also included because a **FULL OUTER JOIN** was used. However, you will notice that the `supplier_id` and `supplier_name` field for those records contain a <null> value.

Exercises

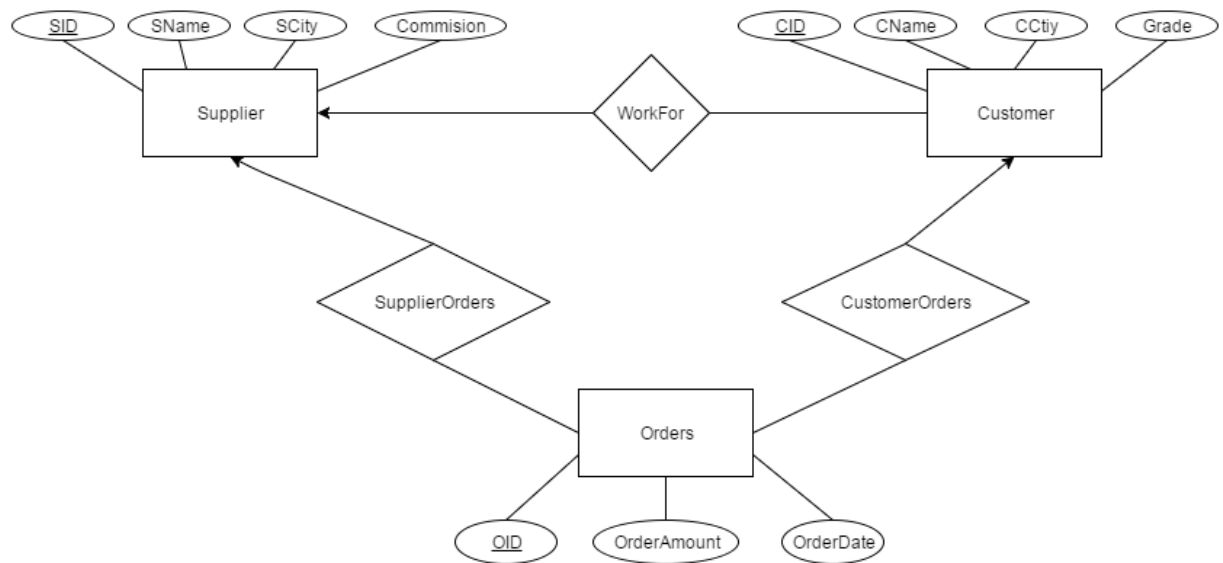


Fig.1. Entity-Relationship Diagram of db_Exp6

- 1) Construct a database named as 'db_Exp6' and create the tables given in Fig. 1.
- 2) Retrive the supplier and customer name and city information who lives in the same city.
- 3) Retrieve customer name, customer city and order amount according to order amount between 10 and 20.
- 4) Write a SQL query that retrieve a list that contains which supplier are working for which customer and obtains a commision from the enterprise is greater than 15%.
- 5) Retrive the name and city of customers who works by own or for a Supplier and has a grade less than 150.
- 6) Retrieve Cname, CCity, OrderDate, OrderAmount, SName, Scity, and commission information that any customers have ordered one or more orders by their supplier or by own.
- 7) Retrieve supplier name and city information who works any (one, more or no) customer and any of the customers who placed either any supplier (one, more or no).
- 8) Write a stored procedure that retrieve customer and order information who placed any order (one or more) or any orders have been placed by the customer who is not on the database.