

# JOINS

## Joins and Keys

Sometimes we have to select data from two or more tables to make our result complete. We have to perform a JOIN. For example, linking companies to contacts in the CRM database.

Tables in a database can be related to each other with keys or ids. A primary key is a column with a unique value for each row. The purpose is to bind data together, across tables, without repeating all of the data in every table.

In the *sales\_company* table, the *companyid* is the primary key, meaning that **no** two rows can have the same *companyid*. The *companyid* distinguishes two persons even if they have the same name.

Look carefully at the *sales\_company* and *sales\_contacts* tables. Notice that:

- The *companyid* column is the primary key of the *sales\_company* table
- The *contactid* column is the primary key of the *sales\_contacts* table
- The *companyid* column in the *sales\_contacts* table is used to refer to the companies on the *sales\_company* table without using their names

## Referring to Two Tables

We can select data from the two tables by referring to two tables, like this:

Link the *sales\_company* table to the *sales\_contacts* table, i.e. connecting contacts to their associated company.

```
SELECT sales_company.company_name ,
sales_contacts.first_name FROM sales_company ,
sales_contacts WHERE sales_company.companyid =
sales_contacts.companyid
```

## Using Joins

OR we can select data from two tables with the JOIN keyword, like this:

```
SELECT sales_company.company_name ,
sales_contacts.first_name FROM sales_company INNER JOIN
sales_contacts ON sales_company.companyid =
sales_contacts.companyid
```

The INNER JOIN returns all rows from both tables where there is a match. If there are no rows in the *sales\_company* that do not have matches in the *sales\_contacts*, those rows will NOT be listed.

# JOINS

## LEFT JOIN

List all companies and their contacts, if any:

```
SELECT sales_company.company_name ,
sales_contacts.first_name FROM sales_company LEFT JOIN
sales_contacts ON sales_company.companyid =
sales_contacts.companyid
```

The LEFT JOIN returns all the rows from the first table (on the LEFT side on the JOIN statement which is *sales\_company*), even if there are no matches in the second table (*sales\_contacts*). If there are rows in the *sales\_company* table that do not have matches in the *sales\_contacts* table, those rows ALSO will be listed.

## RIGHT JOIN

List all contacts, and their companies, if any (note: this query should never have to be used in the CRM system as contacts should ALWAYS have a company):

```
SELECT sales_company.company_name ,
sales_contacts.first_name FROM sales_company RIGHT JOIN
sales_contacts ON sales_company.companyid =
sales_contacts.companyid
```

The RIGHT JOIN returns all the rows from the second table (the one on the right side of the JOIN statement which is *sales\_contacts*), even if there are no matches in the first table (*sales\_company*). If there had been any rows in the *sales\_contacts* table that did not have matches in the *sales\_company* table, those rows ALSO would have been listed.