JOINS & Cats

Let’s imagine array of people and array of cats. People (unfortunately not all of them) live into their own houses, i.e. have addresses. Some cats can be lucky and live with owners and have address too, other are free and walk on their own.

Let’s write them down in the tables.

Array A of People:

|  |  |  |  |
| --- | --- | --- | --- |
| People.id | Firstname | Surname | People.address |
| 1 | Alica | Prett | 100 |
| 2 | Nic | Pretty | 101 |
| 3 | Kevin | Smith | 103 |
| 4 | Maria | Matt | 104 |
| 5 | Alina | Perv | 105 |
| 6 | Olesya | Popova | 106 |
| 7 | Michel | Grabb | 107 |
| 8 | Ivan | Sidorov | 108 |
| 9 | Sandra | Grabb | 107 |
| 10 | Bill | Lee | Null |

Array B of Pets:

|  |  |  |
| --- | --- | --- |
| Pets.id | alias | Pets.address |
| 1 | Lucky | 101 |
| 2 | Marsik | 101 |
| 3 | Barsik | Null |
| 4 | Null | Null |
| 5 | Murka | 104 |
| 6 | Blacky | 106 |
| 7 | Snowy | 107 |
| 8 | Null | 102 |

1. What will we learn if simply connect two tables on address?

(By the way, simple JOIN means INNER JOIN)

|  |
| --- |
| select \* from people A  join pets B – equivalent inner join  on A.address = B.address |
| Diagram, venn diagram  Description automatically generated |
| We answered the question: who live together, at the same addresses?  This is intersection of two sets.  If one people have several animals, he will be mentioned several times. Like an animal with two owners, it will be listed twice, with each owner once. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | 2 | Nic | Pretty | 101 | 1 | Lucky | 101 |
| 2 | 2 | Nic | Pretty | 101 | 2 | Marsik | 101 |
| 3 | 4 | Maria | Matt | 104 | 5 | Murka | 104 |
| 4 | 6 | Olesya | Popova | 106 | 6 | Blacky | 106 |
| 5 | 7 | Michel | Grabb | 107 | 7 | Snowy | 107 |
| 6 | 9 | Sandra | Grabb | 107 | 7 | Snowy | 107 |

1. Now add word LEFT before JOIN:

|  |
| --- |
| select \* from people A  left join pets B  ON A.address = B.address |
| Diagram, venn diagram  Description automatically generated |
| LEFT means that all records from left table (People) will be included in the selection, but the records of the right table (Pets) - as it turns out. If they match the key (address), they will hit, and we will get information about ALL people: whether they have cats or not, we can also see how many cats each owner has.  Let's pay attention to the fact that a person without an address also got into the list, because the table of people is the main one, the table of cats joins it. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | 1 | Alica | Prett | 100 | Null | Null | Null |
| 2 | 2 | Nic | Pretty | 101 | 1 | Lucky | 101 |
| 3 | 2 | Nic | Pretty | 101 | 2 | Marsik | 101 |
| 4 | 3 | Kevin | Smith | 103 | Null | Null | Null |
| 5 | 4 | Maria | Matt | 104 | 5 | Murka | 104 |
| 6 | 5 | Alina | Perv | 105 | Null | Null | Null |
| 7 | 6 | Olesya | Popova | 106 | 6 | Blacky | 106 |
| 8 | 7 | Michel | Grabb | 107 | 7 | Snowy | 107 |
| 9 | 9 | Ivan | Sidorov | 108 | Null | Null | Null |
| 10 | 9 | Sandra | Grabb | 107 | 7 | Snowy | 107 |
| 11 | 10 | Bill | Lee | Null | Null | Null | Null |

1. And if we put RIGTH instead of LEFT and don’t change anything else, the Cat-table become a “main” table:

|  |
| --- |
| select \* from people A  right join pets B  ON A.address = B.address |
| Diagram, venn diagram  Description automatically generated |
| Here we will see ALL cats and their owners (if the owners exist).  If a cat has two or more owners, then the cat will be in the table as many times as it has owners (you need to know all of them).  And of course, cats without an address are on the list too, because now the table of cats is the main one and people join it.  It should be noted that even cats without a name at all (Null in the alias field) were also included in the selection, because the table of cats is the main one here! |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | 2 | Nic | Pretty | 101 | 1 | Lucky | 101 |
| 2 | 2 | Nic | Pretty | 101 | 2 | Marsik | 101 |
| 3 | Null | Null | Null | Null | 3 | Barsik | Null |
| 4 | Null | Null | Null | Null | 4 | Null | Null |
| 5 | 4 | Maria | Matt | 104 | 5 | Murka | 104 |
| 6 | 6 | Olesya | Popova | 106 | 6 | Blacky | 106 |
| 7 | 7 | Michel | Grabb | 107 | 7 | Snowy | 107 |
| 8 | 9 | Sandra | Grabb | 107 | 7 | Snowy | 107 |
| 9 | Null | Null | Null | Null | 8 | Null | 102 |

1. Not all people love cats. Let’s see how many such a person do we have:

|  |
| --- |
| select \* from people A  left join pets B  ON A.address = B.address  where b.address is NULL |
| Diagram, venn diagram  Description automatically generated |
| Select all people, (LEFT JOIN, People is left), then leave those who have NULL in field pets.id: |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | 1 | Alica | Prett | 100 | Null | Null | Null |
| 2 | 3 | Kevin | Smith | 103 | Null | Null | Null |
| 3 | 5 | Alina | Perv | 105 | Null | Null | Null |
| 4 | 9 | Ivan | Sidorov | 108 | Null | Null | Null |
| 5 | 10 | Bill | Lee | Null | Null | Null | Null |

1. Not every pet is so lucky to have his owner:

|  |
| --- |
| select \* from people A  right join pets B  ON A.address = B.address  where A.address is NULL |
| Diagram, logo  Description automatically generated with medium confidence |
| There is a list of free cats.  Even if a cat has a house address, it does not always indicate the presence of a person there. Cats without a alias are also taken into account, because here the table of cats is the main one. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | Null | Null | Null | Null | 3 | Barsik | Null |
| 2 | Null | Null | Null | Null | 4 | Null | Null |
| 3 | Null | Null | Null | Null | 8 | Null | 102 |

1. Suppose we want to arrange a population census, which will include both people and cats, but taking into account residential addresses:

|  |
| --- |
| select \* from people A  full outer join pets B  ON A.address = B.address |
| Diagram, venn diagram  Description automatically generated |
| Сюда попадут люди без котиков, хозяева и их котики, проживающие вместе, котики без хозяев. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | 2 | Nic | Pretty | 101 | 1 | Lucky | 101 |
| 2 | 2 | Nic | Pretty | 101 | 2 | Marsik | 101 |
| 3 | 4 | Maria | Matt | 104 | 5 | Murka | 104 |
| 4 | 6 | Olesya | Popova | 106 | 6 | Blacky | 106 |
| 5 | 7 | Michel | Grabb | 107 | 7 | Snowy | 107 |
| 6 | 9 | Sandra | Grabb | 107 | 7 | Snowy | 107 |
| 7 | 1 | Alica | Prett | 100 | Null | Null | Null |
| 8 | 3 | Kevin | Smith | 103 | Null | Null | Null |
| 9 | 5 | Alina | Perv | 105 | Null | Null | Null |
| 10 | 9 | Ivan | Sidorov | 108 | Null | Null | Null |
| 11 | 10 | Bill | Lee | Null | Null | Null | Null |
| 12 | Null | Null | Null | Null | 3 | Barsik | Null |
| 13 | Null | Null | Null | Null | 4 | Null | Null |
| 14 | Null | Null | Null | Null | 8 | Null | 102 |

1. Let's find all the lonely people: people who don't like cats and cats, who haven't found (or don't love?) people:

|  |
| --- |
| select \* from people A  full outer join pets B  on A.address = B.address  where A.address is NULL or B.address is NULL |
| Diagram  Description automatically generated |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Row number | People.id | Firstname | Surname | People.address | Pets.id | alias | Pets.address |
| 1 | 1 | Alica | Prett | 100 | Null | Null | Null |
| 2 | 3 | Kevin | Smith | 103 | Null | Null | Null |
| 3 | 5 | Alina | Perv | 105 | Null | Null | Null |
| 4 | 9 | Ivan | Sidorov | 108 | Null | Null | Null |
| 5 | 10 | Bill | Lee | Null | Null | Null | Null |
| 6 | Null | Null | Null | Null | 3 | Barsik | Null |
| 7 | Null | Null | Null | Null | 4 | Null | Null |
| 8 | Null | Null | Null | Null | 8 | Null | 102 |

1. And now imagine that each person wanted to treat each cat, regardless of the address. How many bags of food will be used?

|  |
| --- |
| select \* from people A  cross join pets B |
| This is a Cartesian product - every member of one table meets every member of another. To calculate the number of food packets, you need to multiply the number of records in one table by the number of records in another. In our case, we need 10x8=80 bags. |
| \*By the way, in January 2022, a law came into force in Spain, according to which pet owners are no longer called owners, but guardians. |