任务 6-2 连接语句

目录

Contents

- Hive Join语法规则
- Hive 6种Join方式详解
- Hive Join使用注意事项

Join概念回顾

- 根据数据库的三范式设计要求和日常工作习惯来说，我们通常不会设计一张大表把所有类型的数据都放在一起，而是不同类型的数据设计不同的表存储。
- 比如在设计一个订单数据表的时候，可以将客户编号作为一个外键和订单表建立相应的关系。而不可以在订单表中添加关于客户其它信息（比如姓名、所属公司等）的字段。

订单信息表

订单编号	订单项目	负责人	业务员	订单数量	客户编号
001	挖掘机	刘明	李东明	1台	1
002	冲击钻	李刚	霍新峰	8个	2
003	铲车	郭新一	艾美丽	2辆	1

客户信息表

客户编号	客户名称	所属公司	联系方式
1	李聪	五一建设	13253661015
2	刘新明	个体经营	13285746958

Join概念回顾

- 在这种情况下，有时需要基于多张表查询才能得到最终完整的结果；
- join语法的出现是用于根据两个或多个表中的列之间的关系，从这些表中共同组合查询数据。

订单信息表

订单编号	订单项目	负责人	业务员	订单数量	客户编号
001	挖掘机	刘明	李东明	1台	1
002	冲击钻	李刚	霍新峰	8个	2
003	铲车	郭新一	艾美丽	2辆	1

客户信息表

客户编号	客户名称	所属公司	联系方式
1	李聪	五一建设	13253661015
2	刘新明	个体经营	13285746958

Join语法规则

- 在Hive中，当下版本3.1.2总共支持6种join语法。分别是：
  - inner join（内连接）、
  - left join（左连接）、
  - right join（右连接）
  - full outer join（全外连接）、
  - left semi join（左半开连接）、
  - cross join（交叉连接，也叫做笛卡尔乘积（cartesian product））。

Join语法规则

- 在Hive中，当下版本3.1.2总共支持6种join语法。分别是：
  - inner join（内连接）、
  - left join（左连接）、
  - right join（右连接）
  - full outer join（全外连接）、
  - left semi join（左半开连接）、
  - cross join（交叉连接，也叫做笛卡尔乘积（cartesian product））。

Join语法规则

- table_reference：是 join 查询中使用的表名，也可以是子查询别名（查询结果当成表参与 join）。
- table_factor：与 table_reference 相同，是联接查询中使用的表名，也可以是子查询别名。
- join_condition：join 查询关联的条件，如果在两个以上的表上需要连接，则使用 AND 关键字。

```
join_table:
    table_reference [INNER] JOIN table_factor [join_condition]
  | table_reference {LEFT|RIGHT|FULL} [OUTER] JOIN table_reference join_condition
  | table_reference LEFT SEMI JOIN table_reference join_condition
  | table_reference CROSS JOIN table_reference [join_condition] (as of Hive 0.10)

join_condition:
    ON expression
```

Join语法丰富化

- Hive中的join语法从面世开始其实并不丰富，不像在RDBMS中那么灵活。
- 从Hive 0.13.0开始，支持隐式联接表示法（请参阅HIVE-5558）。允许FROM子句连接以逗号分隔的表列表，而省略JOIN关键字。
- 从Hive 2.2.0开始，支持ON子句中的复杂表达式，支持不相等连接（请参阅HIVE-15211和HIVE-15251）。在此之前，Hive不支持不是相等条件的联接条件。

```
SELECT *
FROM table1 t1, table2 t2, table3 t3
WHERE t1.id = t2.id AND t2.id = t3.id AND t1.zipcode = '02535';

SELECT a.* FROM a JOIN b ON (a.id = b.id)
SELECT a.* FROM a JOIN b ON (a.id = b.id AND a.department =
b.department)
SELECT a.* FROM a LEFT OUTER JOIN b ON (a.id <> b.id)
```

Join查询数据环境准备

- 为了更好的练习、学习掌握Hive中的join语法， 下面我们去创建3张表并且加载数据到表中。
  - ✓ 表1： employee员工表
  - ✓ 表2： employee_address员工住址信息表;
  - ✓ 表3： employee_connection 员工联系方式表;

```
-- table1: 员工表
CREATE TABLE employee(
    id int,
    name string,
    deg string,
    salary int,
    dept string
) row format delimited
fields terminated by ',';
```

```
-- table2: 员工家庭住址信息表
CREATE TABLE employee_address (
    id int,
    hno string,
    street string,
    city string
) row format delimited
fields terminated by ',';
```

```
-- table3: 员工联系方式信息表
CREATE TABLE employee_connection (
    id int,
    phno string,
    email string
) row format delimited
fields terminated by ',';
```

Join查询数据环境准备

```
load data local inpath '/usr/datas/employee.txt'
into table employee;
select * from employee;

load data local inpath '/usr/datas/employee_address.txt'
into table employee_address;
select * from employee_address;

load data local inpath '/usr/datas/employee_connection.txt'
into table employee_connection;
select * from employee_connection;
```

```
select *
from employee;
```

	id	name	deg	salary	dept
1	1201	gopal	manager	50000	TP
2	1202	manisha	cto	50000	TP
3	1203	khalil	dev	30000	AC
4	1204	prasanth	dev	30000	AC
5	1206	kranthi	admin	20000	TP

```
select *
from employee_address;
```

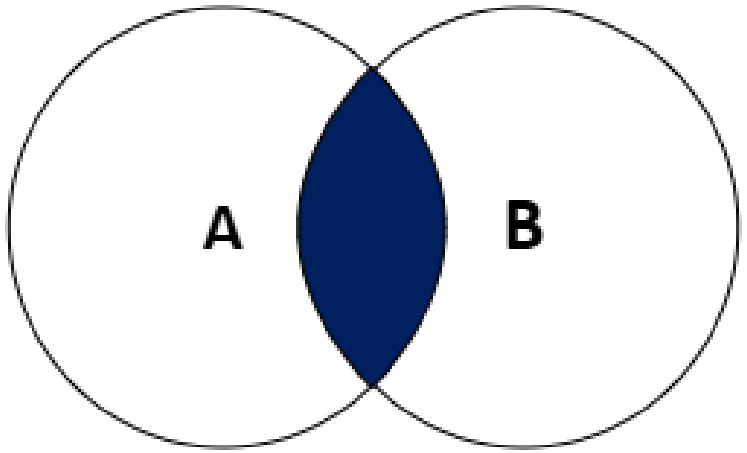
	id	hno	street	city
1	1201	288A	vgiri	jublee
2	1202	108I	aoc	ny
3	1204	144Z	pgutta	hyd
4	1206	78B	old city	la
5	1207	720X	hitec	ny

```
select *
from employee_connection;
```

	id	phno	email
1	1201	2356742	gopal@tp.com
2	1203	1661663	manisha@tp.com
3	1204	8887776	khalil@ac.com
4	1205	9988774	prasanth@ac.com
5	1206	1231231	kranthi@tp.com

inner join 内连接

- 内连接是最常见的一种连接，它也被称为普通连接，其中inner可以省略：**inner join == join**;
- 只有进行连接的两个表中都存在与连接条件相匹配的数据才会被留下来。



6.2.1 Hive SQL Join连接操作

inner join 内连接

```
-- 1、inner join
select e.id, e.name, e_a.city, e_a.street
from employee e inner join employee_address e_a
on e.id = e a.id;
```

id	name	deg	salary	dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1203	khalil	dev	30000	AC
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

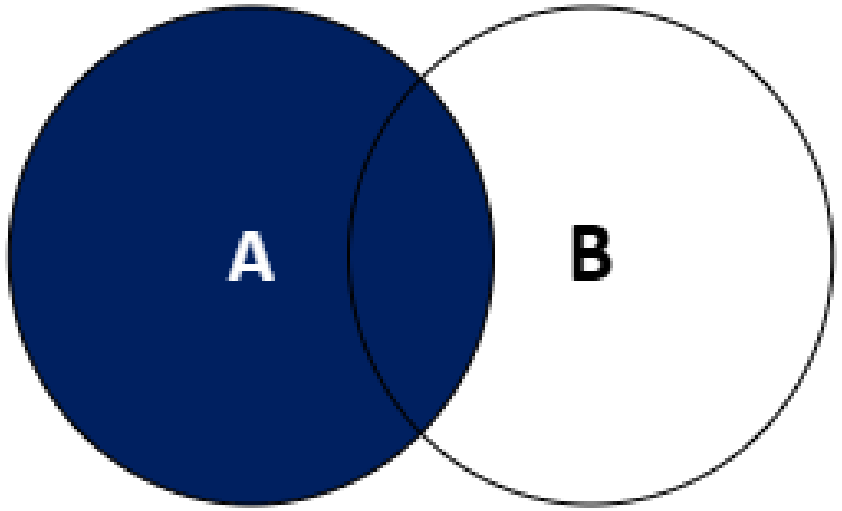
id	hno	street	city
1201	288A	vgiri	jubilee
1202	108I	aoc	ny
1204	144Z	pgutta	hyd
1206	78B	old city	la
1207	720X	hitec	ny

Inner join

e.id	e.name	e_a.city	e_a.street
1201	gopal	jubilee	vgiri
1202	manisha	ny	aoc
1204	prasanth	hyd	pgutta
1206	kranthi	la	old city

left join 内连接

- left join 中文叫做是 **左外连接** (Left Outer Join) 或者左连接，其中 outer 可以省略，left outer join 是早期的写法。
- left join 的核心就在于 left 左。左指的是 join 关键字左边的表，简称左表。
- 通俗解释：join时以左表的全部数据为准，右边与之关联；左表数据全部返回，右表关联上的显示返回，关联不上的显示 null 返回。



6.2.1 Hive SQL Join连接操作

left join 内连接

```
-- 2、left join
select e.id, e.name, e_conn.phno, e_conn.email
from employee e left join employee_connection e_conn
on e.id = e_conn.id;
```

id	name	deg	salary	dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1203	khalil	dev	30000	AC
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

employee_connection.id	employee_connection.phno	employee_connection.email
1201	2356742	gopal@tp.com
1203	1661663	manisha@tp.com
1204	8887776	khalil@ac.com
1205	9988774	prasanth@ac.com
1206	1231231	krathii@tp.com

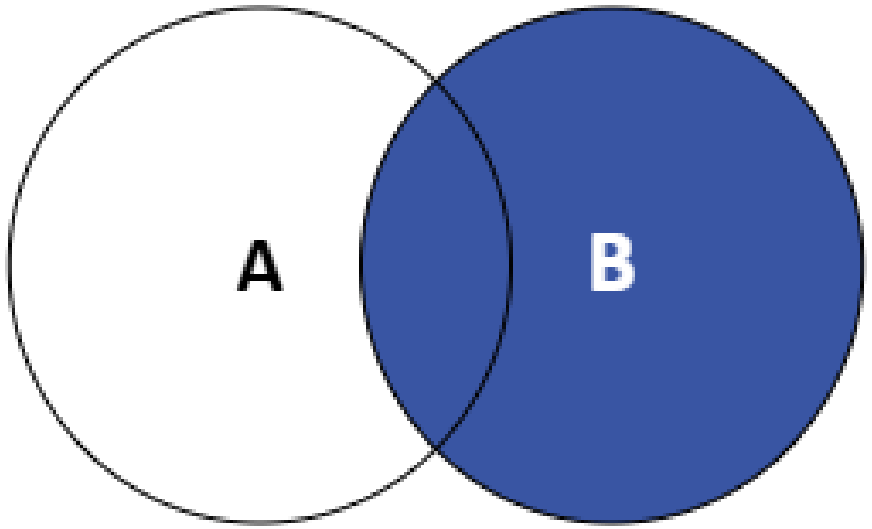
left join

e.id	e.name	e_conn.phno	e_conn.email
1201	gopal	2356742	gopal@tp.com
1202	manisha	NULL	NULL
1203	khalil	1661663	manisha@tp.com
1204	prasanth	8887776	khalil@ac.com
1206	kranthi	1231231	krathii@tp.com

没有join上的显示null

right join 内连接

- left join 中文叫做是 **左外连接** (Left Outer Join) 或者左连接，其中 outer 可以省略，left outer join 是早期的写法。
- left join 的核心就在于 left 左。左指的是 join 关键字左边的表，简称左表。
- 通俗解释：join时以左表的全部数据为准，右边与之关联；左表数据全部返回，右表关联上的显示返回，关联不上的显示 null 返回。
- 很明显， right join和left join之间很相似，重点在于以哪边为准，也就是一个方向的问题。



6.2.1 Hive SQL Join连接操作

right join 内连接

```
-- 2、left join
select e.id, e.name, e_conn.phno, e_conn.email
from employee e left join employee_connection e_conn
on e.id = e_conn.id;
```

id	name	deg	salary	dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1203	khalil	dev	30000	AC
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

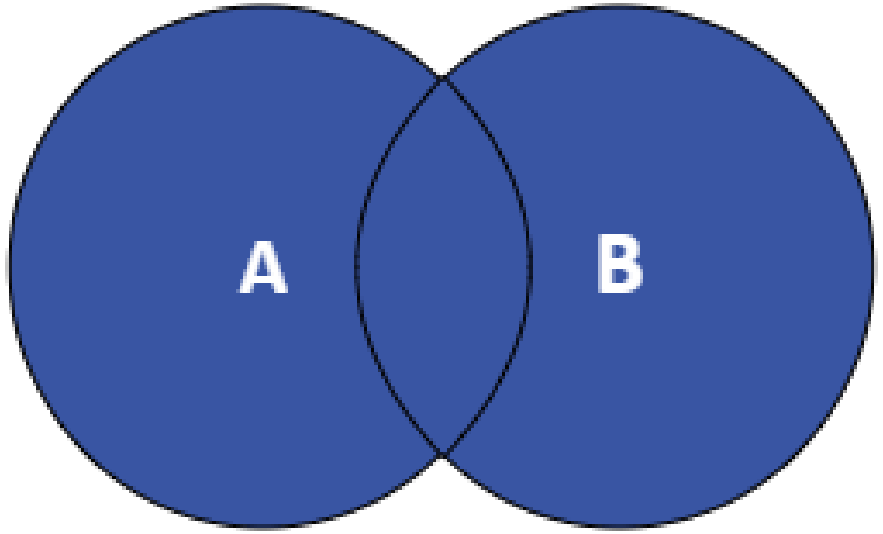
employee_connection.id	employee_connection.phno	employee_connection.email
1201	2356742	gopal@tp.com
1203	1661663	manisha@tp.com
1204	8887776	khalil@ac.com
1205	9988774	prasanth@ac.com
1206	1231231	krathii@tp.com

left join

e.id	e.name	e_conn.phno	e_conn.email
1201	gopal	2356742	gopal@tp.com
1203	khalil	1661663	manisha@tp.com
1204	prasanth	8887776	khalil@ac.com
NULL	NULL	9988774	prasanth@ac.com
1206	kranthi	1231231	krathii@tp.com

full outer join 内连接

- full outer join 等价 full join，中文叫做**全外连接或者外连接**。
- 包含左、右两个表的全部行，不管另外一边的表中是否存在与它们匹配的行;
- 在功能上：等价于对这两个数据集合分别进行左外连接和右外连接，然后再使用消去重复行的操作将上述两个结果集合并为一个结果集。



6.2.1 Hive SQL Join连接操作

full outer join 内连接

```
-- 4、full outer join
select e.id, e.name, e_a.city, e_a.street
from employee e full outer join employee_address e_a
on e.id = e_a.id;
```

id	name	deg	salary	dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1203	khalil	dev	30000	AC
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

id	hno	street	city
1201	288A	vgiri	jubilee
1202	108I	aoc	ny
1204	144Z	pgutta	hyd
1206	78B	old city	la
1207	720X	hitec	ny

left join

e.id	e.name	e_a.city	e_a.street
1201	gopal	jubilee	vgiri
1202	manisha	ny	aoc
1203	khalil	NULL	NULL
1204	prasanth	hyd	pgutta
1206	kranthi	la	old city
NULL	NULL	ny	hitec

left semi join 内连接

- 左半开连接（LEFT SEMI JOIN)会返回左边表的记录，前提是记录对右边的表满足ON语句中的判定条件。
- 从效果上来看有点像inner join之后只返回左表的结果。

工作原理:

- 在右表中仅确认是否存在（类似于 EXISTS 子查询）
- 左表的行只要在右表中**有任意一个匹配**就会被选择
- **不返回右表的数据**

6.2.1 Hive SQL Join连接操作

left semi join 内连接

```
-- 5, left semi join
select *
from employee e left semi join employee_address e_addr
on e.id = e_addr.id;
```

id	name	deg	salary	dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1203	khalil	dev	30000	AC
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

id	hno	street	city
1201	288A	vgiri	jubilee
1202	108I	aoc	ny
1204	144Z	pgutta	hyd
1206	78B	old city	la
1207	720X	hitec	ny

Left semi join

e.id	e.name	e.deg	e.salary	e.dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

cross join 交叉连接

- **交叉连接** cross join，将会返回被连接的两个表的笛卡尔积(cartesian product, decarte product)，返回结果的行数等于两个表行数的乘积。对于大表来说，cross join 慎用。
- 在 SQL 标准中定义的 cross join 就是无条件的 inner join。返回两个表的笛卡尔积，无需指定关联键。
- 在 HiveSQL 语法中，cross join 后面可以跟 where 子句进行过滤，或者 on 条件过滤

id	name	deg	salary	dept
1201	gopal	manager	50000	TP
1202	manisha	cto	50000	TP
1203	khalil	dev	30000	AC
1204	prasanth	dev	30000	AC
1206	kranthi	admin	20000	TP

id	hno	street	city
1201	288A	vgiri	jubilee
1202	108I	aoc	ny
1204	144Z	pgutta	hyd
1206	78B	old city	la
1207	720X	hitec	ny

cross join

a.id	a.name	a.deg	a.salary	a.dept	b.id	b.hno	b.street	b.city
1201	gopal	manager	50000	TP	1201	288A	vgiri	jubilee
1202	manisha	cto	50000	TP	1202	108I	aoc	ny
1204	prasanth	dev	30000	AC	1204	144Z	pgutta	hyd
1206	kranthi	admin	20000	TP	1206	78B	old city	la

感谢您的观看



《Hive数据仓库应用教程》