



El empleo
es de todos

Mintrabajo

DESARROLLO DE BASE DE DATOS CON MYSQL



@SENAcomunica

www.sena.edu.co



DESARROLLO DE BASE DE DATOS CON MYSQL



@SENAcomunica

www.sena.edu.co

5

VISUALIZACION DE DATOS DE VARIAS TABLAS

Después de completar esta lección usted estará en la capacidad de:

- **Escribir sentencias SELECT para acceder a datos de más de una tabla mediante uniones igualitarias y no igualitarias.**
- **Unir una tabla así misma mediante una auto unión.**
- **Ver datos que no cumple con los criterios de unión utilizando uniones externas.**
- **Generar producto cartesiano de todas las filas de dos o más tablas.**

Tipos de uniones



- **Uniones cruzadas**
- **Uniones naturales**
- **Cláusula USING**
- **Uniones externas completas(o de dos lados)**
- **Condiciones de unión arbitrarias para uniones externas**

Creación de uniones naturales



- La cláusula **NATURAL JOIN** se basa en todas las columnas de las dos tablas que tienen el mismo nombre.
- Selecciona filas de las dos tablas que tienen valores iguales en todas las columnas correspondientes.
- Si las columnas tienen los mismos nombres y tipo de datos diferentes se genera un error.

Creación de uniones naturales



```
MariaDB [HR]> SELECT DEPARTMENT_ID,DEPARTMENT_NAME,  
-> LOCATION_ID,CITY  
-> FROM DEPARTMENTS  
-> NATURAL JOIN LOCATIONS;
```

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
10	Administration	1700	Seattle
20	Marketing	1800	Toronto
30	Purchasing	1700	Seattle
40	Human Resources	2400	London
50	Shipping	1500	South San Francisco
60	IT	1400	Southlake
70	Public Relations	2700	Munich
80	Sales	2500	Oxford

...

240	Government Sales	1700	Seattle
250	Retail Sales	1700	Seattle
260	Recruiting	1700	Seattle
270	Payroll	1700	Seattle
280	SIN DATOS	3200	Mexico City

```
28 rows in set (0.09 sec)
```

Creación de uniones con la clausula USING



- Si hay varias columnas que tienen los mismos nombres pero los tipos de datos no corresponden, se puede utilizar la cláusula USING para especificar la columna para realizar la unión.
- Utilice la cláusula USING para asignar sólo una columna cuando corresponde a más de una columna.
- No utilice un alias en la tabla o columnas a las que se hace referencia.
- Las cláusulas NATURAL JOIN y USING se excluyen mutuamente.

Creación de uniones naturales



```
MariaDB [HR]> SELECT EMPLOYEE_ID, LAST_NAME, LOCATION_ID, DEPARTMENT_ID  
-> FROM EMPLOYEES  
-> JOIN DEPARTMENTS  
-> USING(DEPARTMENT_ID);
```

EMPLOYEE_ID	LAST_NAME	LOCATION_ID	DEPARTMENT_ID
103	Hunold	1400	60
104	Ernst	1400	60
105	Austin	1400	60
106	Pataballa	1400	60
107	Lorentz	1400	60

...

172	Bates	2500	80
173	Kumar	2500	80
174	Abel	2500	80
175	Hutton	2500	80
176	Taylor	2500	80
177	Livingston	2500	80
179	Johnson	2500	80
204	Baer	2700	70

106 rows in set (0.01 sec)

Cualificación de nombre de columna ambigua



- **Utilice prefijos de tabla para cualificar nombres de columna que estén varias tablas.**
- **Use prefijos de tablas para mejorar el rendimiento.**
- **Utilice alias de columnas para distinguir columna que tengan nombres idénticos pero que estén en tablas diferentes.**
- **No utilice alias de columnas que estén identificadas en cláusulas USING y que se muestren en cualquier otra parte de la sentencia SQL.**

Cualificación de nombre de columna ambigua



```
MariaDB [HR]> SELECT EMPLOYEE_ID, LAST_NAME,  
-> LOCATION_ID, D.DEPARTMENT_ID  
-> FROM EMPLOYEES JOIN DEPARTMENTS D  
-> USING(DEPARTMENT_ID);
```

EMPLOYEE_ID	LAST_NAME	LOCATION_ID	DEPARTMENT_ID
103	Hunold	1400	60
104	Ernst	1400	60
105	Austin	1400	60
106	Pataballa	1400	60
107	Lorentz	1400	60
120	Weiss	1500	50
121	Fripp	1500	50
122	Kaufling	1500	50

...

176	Taylor	2500	80
177	Livingston	2500	80
179	Johnson	2500	80
204	Baer	2700	70

```
106 rows in set (0.00 sec)
```

Creación de uniones con la cláusula ON



- La condición de union para la unión natural es básicamente una unión igualitaria de todas las columnas con el mismo nombre.
- Utilice la cláusula ON para especificar condiciones arbitrarias o para especificar las columnas que se unirán.
- La condición de unión se separa de otras condiciones de búsqueda.
- la cláusula ON facilita la comprensión del código.

Recuperación de filas con la clausula ON



```
MariaDB [HR]> SELECT EMPLOYEE_ID, LAST_NAME,  
-> LOCATION_ID, D.DEPARTMENT_ID  
-> FROM EMPLOYEES E  
-> JOIN DEPARTMENTS D  
-> ON E.DEPARTMENT_ID=D.DEPARTMENT_ID;
```

EMPLOYEE_ID	LAST_NAME	LOCATION_ID	DEPARTMENT_ID
103	Hunold	1400	60
104	Ernst	1400	60
105	Austin	1400	60
106	Pataballa	1400	60
107	Lorentz	1400	60
120	Weiss	1500	50
121	Fripp	1500	50

...

177	Livingston	2500	80
179	Johnson	2500	80
204	Baer	2700	70

```
106 rows in set (0.00 sec)
```

Aplicación de condiciones adicionales a una unión.



```
MariaDB [HR]> SELECT EMPLOYEE_ID, LAST_NAME,  
-> LOCATION_ID, D.DEPARTMENT_ID  
-> FROM EMPLOYEES E  
-> JOIN DEPARTMENTS D  
-> ON E.DEPARTMENT_ID=D.DEPARTMENT_ID  
-> AND E.MANAGER_ID=149;
```

EMPLOYEE_ID	LAST_NAME	LOCATION_ID	DEPARTMENT_ID
174	Abel	2500	80
175	Hutton	2500	80
176	Taylor	2500	80
177	Livingston	2500	80
179	Johnson	2500	80

```
5 rows in set (0.01 sec)
```


Creación tabla JOB_GRADES.



```
CREATE TABLE job_grades (  
  grade CHAR(1),  
  lowest_sal DECIMAL(8,2) NOT NULL,  
  highest_sal DECIMAL(8,2) NOT NULL  
);  
ALTER TABLE job_grades  
ADD CONSTRAINT jobgrades_grade_pk  
PRIMARY KEY (grade);  
INSERT INTO job_grades VALUES ('A', 1000, 2999);  
INSERT INTO job_grades VALUES ('B', 3000, 5999);  
INSERT INTO job_grades VALUES ('C', 6000, 9999);  
INSERT INTO job_grades VALUES ('D', 10000, 14999);  
INSERT INTO job_grades VALUES ('E', 15000, 24999);  
INSERT INTO job_grades VALUES ('F', 25000, 40000);
```

```
MariaDB [HR]> select * from job_grades;  
+-----+-----+-----+  
| grade | lowest_sal | highest_sal |  
+-----+-----+-----+  
| A     | 1000.00    | 2999.00    |  
| B     | 3000.00    | 5999.00    |  
| C     | 6000.00    | 9999.00    |  
| D     | 10000.00   | 14999.00   |  
| E     | 15000.00   | 24999.00   |  
| F     | 25000.00   | 40000.00   |  
+-----+-----+-----+  
6 rows in set (0.00 sec)
```

Recuperación de filas con uniones no igualitarias



```
MariaDB [HR]> SELECT LAST_NAME, SALARY, GRADE  
-> FROM EMPLOYEES E JOIN JOB_GRADES J  
-> ON E.SALARY BETWEEN LOWEST_SAL AND HIGHEST_SAL;
```

LAST_NAME	SALARY	GRADE
King	24000.00	E
Kochhar	17000.00	E
De Haan	17000.00	E
Hunold	9000.00	C
Ernst	6000.00	C
Austin	4800.00	B
Pataballa	4800.00	B
Lorentz	4200.00	B
Greenberg	12000.00	D

...

Fay	6000.00	C
Mavris	6500.00	C
Baer	10000.00	D
Higgins	12000.00	D
Gietz	8300.00	C

```
+-----+-----+  
107 rows in set (0.01 sec)
```


Uniones INNER frente a OUTER



- En el ANSI SQL:1999, la unión de dos tablas que devuelven solo filas con correspondencia se denomina unión interna.
- Una unión entre dos tablas que devuelven los resultados de la unión interna y las filas sin correspondencia de las tablas a la izquierda (o derecha), se denomina unión externa izquierda o derecha.
- Una unión entre dos tablas que devuelven los resultados de la unión interna y los resultados de una unión izquierda y derecha es una unión externa completa.

LEFT OUTER



```
MariaDB [HR]> SELECT LAST_NAME, D.DEPARTMENT_ID, DEPARTMENT_NAME
-> FROM EMPLOYEES E
-> LEFT OUTER JOIN DEPARTMENTS D
-> ON (E.DEPARTMENT_ID=D.DEPARTMENT_ID);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Kochhar	90	Executive
De Haan	90	Executive
Hunold	60	IT
Ernst	60	IT
Austin	60	IT
Pataballa	60	IT
Lorentz	60	IT
Greenberg	100	Finance
Faviet	100	Finance
Chen	100	Finance
...		
Livingston	80	Sales
Grant	NULL	NULL
Johnson	80	Sales
...		
Baer	70	Public Relations
Higgins	110	Accounting
Gietz	110	Accounting
	NULL	NULL

```
108 rows in set (0.00 sec)
```

RIGHT OUTER JOIN



```
MariaDB [HR]> SELECT LAST_NAME,D.DEPARTMENT_ID,DEPARTMENT_NAME  
-> FROM EMPLOYEES E  
-> RIGHT OUTER JOIN DEPARTMENTS D  
-> ON (E.DEPARTMENT_ID=D.DEPARTMENT_ID);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Hartstein	20	Marketing
Fay	20	Marketing

...

NULL	240	Government Sales
NULL	250	Retail Sales
NULL	260	Recruiting
NULL	270	Payroll
NULL	280	SIN DATOS

```
+-----+-----+  
123 rows in set (0.00 sec)
```



GRACIAS

Línea de atención al ciudadano: 018000 910270
Línea de atención al empresario: 018000 910682



@SENAcomunica

www.sena.edu.co

CRÉDITOS



Realizado por el instructor José Fernando Galindo Suárez
jgalindos@sena.edu.co 2020

