

Инструкция SELECT

Многотабличные запросы

Лекция 8

Доцент каф. ВТиКГ Данилова Е.В.

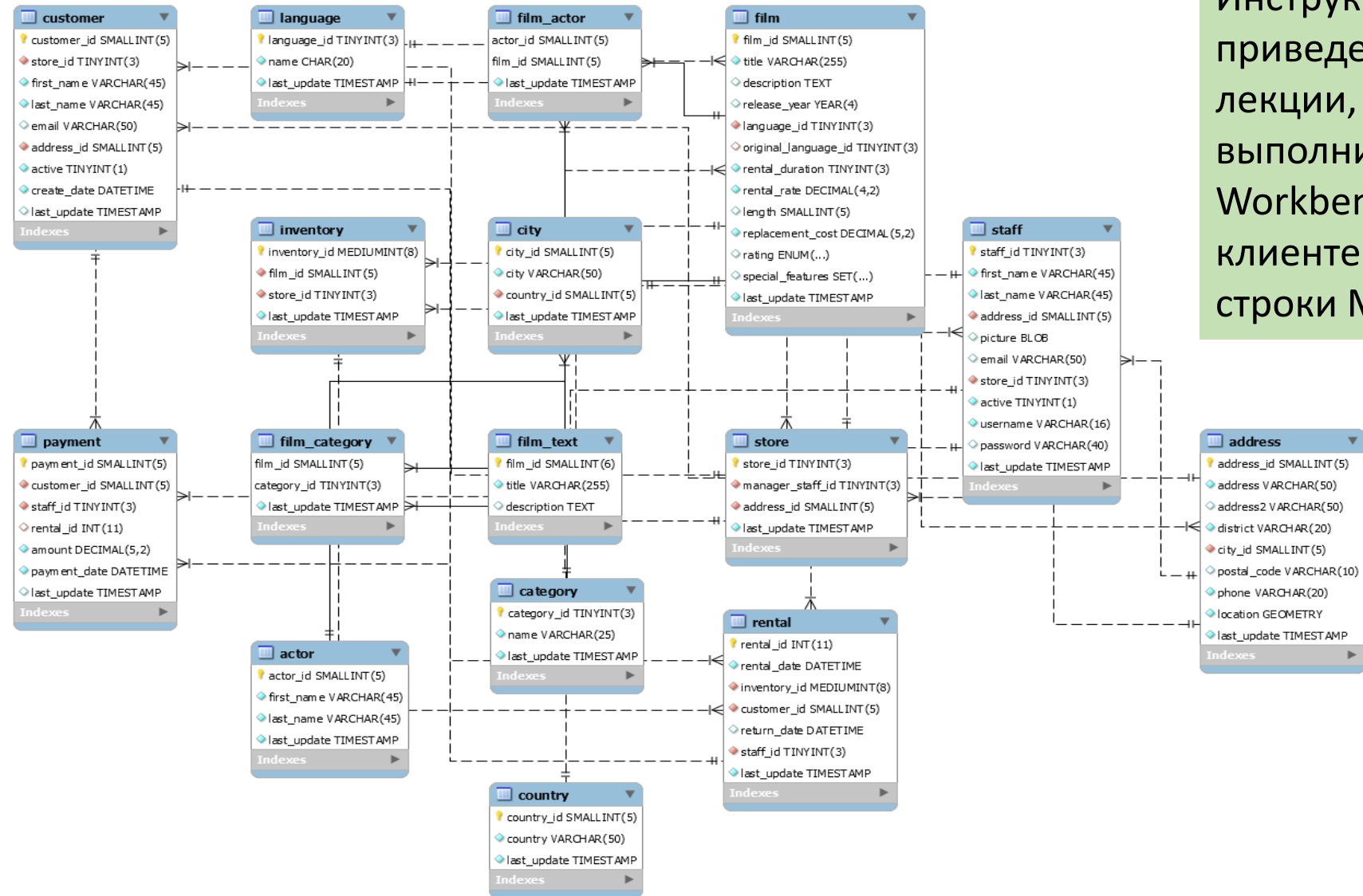
ПЛАН

1. Информация о примерах
2. Назначение многотабличных запросов
3. Простое соединение таблиц
4. Запросы с использованием отношения «предок-потомок»
5. Соединения с разными условиями отбора строк
6. Естественные соединения
7. Запросы к трем и более таблицам
8. Прочие соединения по равенству
9. Соединение по неравенству
10. Особенности многотабличных запросов

Информация о примерах

1

Все примеры созданы с использованием демонстрационной базы данных **Sakila**, которая входит в состав СУБД MySQL



Инструкции SQL, приведенные в лекции, можно выполнить в Workbench или клиенте командной строки MySQL

Рис. 1. EER-диаграмма, база данных Sakila

В таблице **address** хранятся адреса сотрудников, магазинов и клиентов. Сведения о сотрудниках хранятся в таблице **staff**, о магазинах – в таблице **store**, о клиентах – в таблице **customer**

	store_id	manager_staff_id	address_id	last_update
▶	1	1	1	2006-02-15 04:57:12
	2	2	2	2006-02-15 04:57:12
	NULL	NULL	NULL	NULL

Рис. 2. Содержимое таблицы **store**

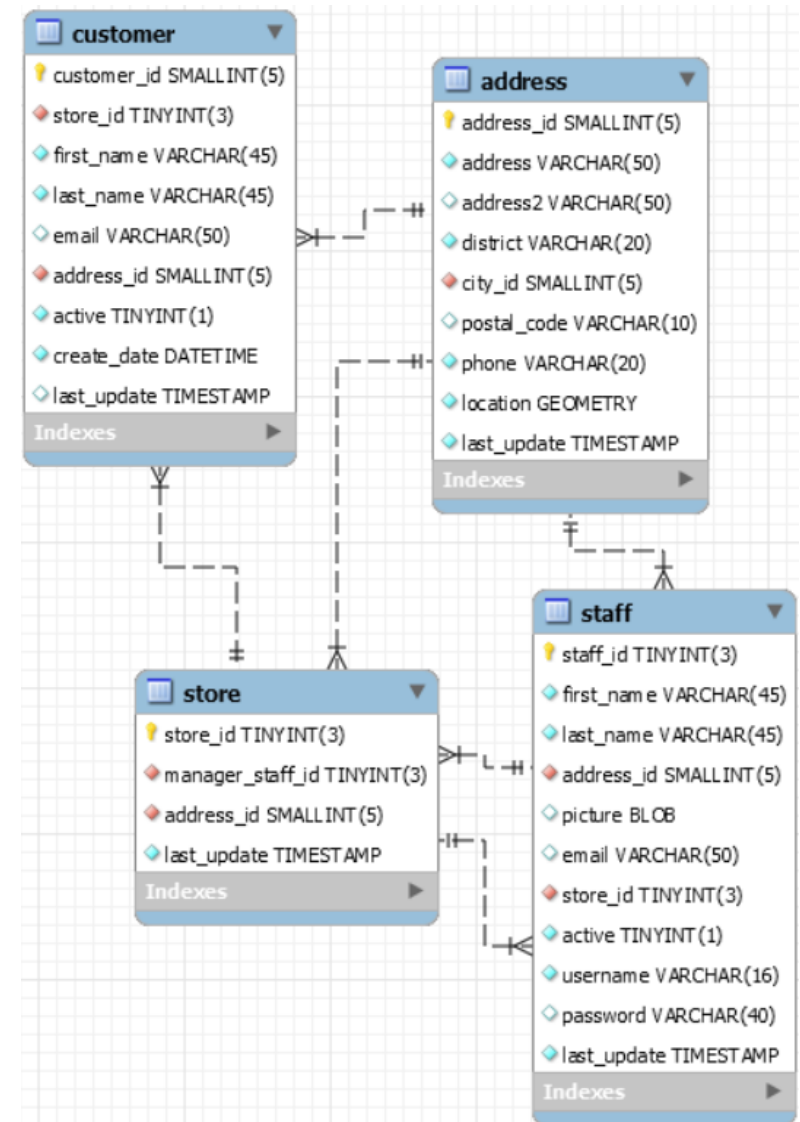


Рис. 4. Фрагмент EER-диаграммы 5

	address_id	address	address2	district	city_id	postal_code	phone	location	last_update
▶	1	47 MySakila Drive	NULL	Alberta	300			BLOB	2014-09-25 22:30:27
	2	28 MySQL Boulevard	NULL	QLD	576			BLOB	2014-09-25 22:30:09
	3	23 Workhaven Lane	NULL	Alberta	300		14033335568	BLOB	2014-09-25 22:30:27
	4	1411 Lillydale Drive	NULL	QLD	576		6172235589	BLOB	2014-09-25 22:30:09
	5	1913 Hanoi Way		Nagasaki	463	35200	28303384290	BLOB	2014-09-25 22:31:53
	6	1121 Loja Avenue		California	449	17886	838635286649	BLOB	2014-09-25 22:34:01
	7	692 Joliet Street		Attika	38	83579	448477190408	BLOB	2014-09-25 22:31:07
	8	1566 Inegl Manor		Mandalay	349	53561	705814003527	BLOB	2014-09-25 22:32:18
	9	53 Idfu Parkway		Nantou	361	42399	10655648674	BLOB	2014-09-25 22:33:16
	10	1795 Santiago de ...		Texas	295	18743	860452626434	BLOB	2014-09-25 22:33:55
	11	900 Santiago de C...		Centra...	280	93896	716571220373	BLOB	2014-09-25 22:34:11
	12	478 Joliet Way		Hamilton	200	77948	657282285970	BLOB	2014-09-25 22:32:22

Рис. 3. Фрагмент содержимого таблицы **address**

	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
▶	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer....	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	5	1	ELIZABETH	BROWN	ELIZABETH.BROWN@sakilacustomer.org	9	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	6	2	JENNIFER	DAVIS	JENNIFER.DAVIS@sakilacustomer.org	10	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	7	1	MARIA	MILLER	MARIA.MILLER@sakilacustomer.org	11	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	8	2	SUSAN	WILSON	SUSAN.WILSON@sakilacustomer.org	12	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	9	2	MARGARET	MOORE	MARGARET.MOORE@sakilacustomer.org	13	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	10	1	DOROTHY	TAYLOR	DOROTHY.TAYLOR@sakilacustomer.org	14	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	11	2	LISA	ANDERSON	LISA.ANDERSON@sakilacustomer.org	15	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	12	1	NANCY	THOMAS	NANCY.THOMAS@sakilacustomer.org	16	1	2006-02-14 22:04:36	2006-02-15 04:57:20

Рис. 5. Фрагмент содержимого таблицы **customer**

	staff_id	first_name	last_name	address_id	picture	email	store_id	active	username	password	last_update
▶	1	Mike	Hillyer	3	BLOB	Mike.Hillyer@sakilastaff.com	1	1	Mike	8cb2237d...	2006-02-15 03:57:16
	2	Jon	Stephens	4	NULL	Jon.Stephens@sakilastaff.com	2	1	Jon	NULL	2006-02-15 03:57:16

Рис. 6. Содержимое таблицы **staff**

Назначение многотабличных запросов

2

- Как правило, работая с клиентским приложением для решения бизнес-задач, ни один пользователь не работает напрямую с таблицами базы данных, не знает внутренней структуры базы данных, не имеет представления о том, что информация, относящаяся к одной сущности, хранится в разных таблицах, которые связаны друг с другом по значениям внешнего ключа.
- Таким образом, большинство запросов, реализующих бизнес-логику приложения, извлекают информацию не из одной, а сразу из нескольких таблиц, то есть являются **многотабличными**.
- **Многотабличные запросы** – запросы, **соединяющие** данные из нескольких таблиц базы данных.

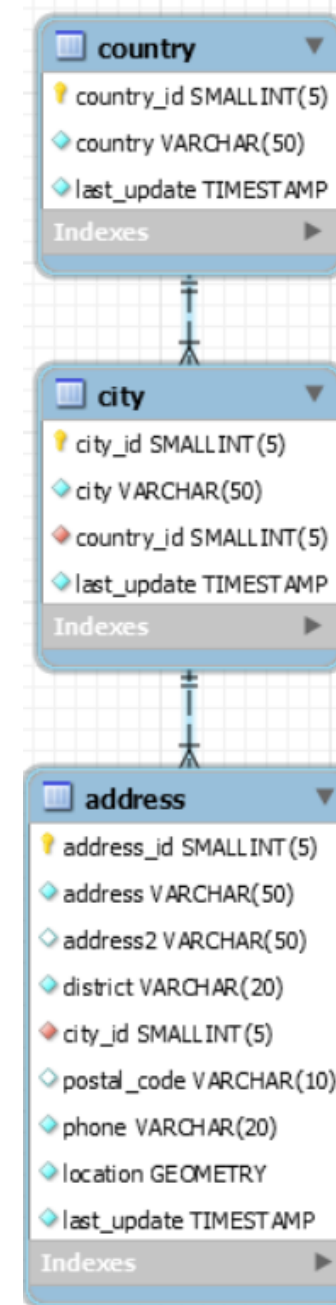


Рис. 1. Сущность «Адрес» – информация хранится в трех таблицах

Простое соединение таблиц

3

- **Соединение таблиц** – процесс формирования пар строк путем сравнения содержимого соответствующих столбцов.
- Получаемая в результате таблица, содержащая данные из обеих исходных таблиц, называется соединением (**JOIN**) двух таблиц.
- Соединение на основе точного равенства между двумя столбцами более корректно именуется соединением по равенству (**EQUI-JOIN**).
- Столбцы из двух различных таблиц, значения которых сравниваются, называются **связанными**, или **согласованными** (matching).



*Одинаковый цвет означает равенство значений

Простой способ соединения таблиц

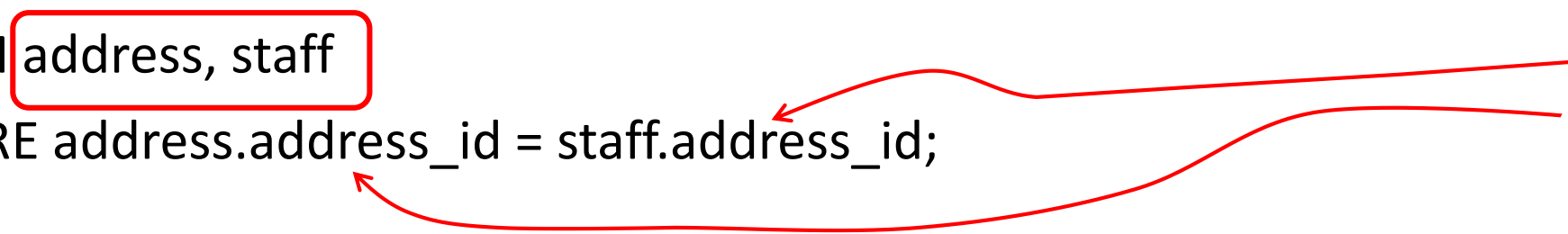
- Простейший способ указать соединяемые таблицы состоит в их перечислении в списке в предложении FROM инструкции SELECT.
- Этот метод указания соединяемых таблиц появился в самых ранних реализациях SQL фирмы IBM. Он включен в исходный SQL-стандарт и поддерживается всеми SQL-базами данных.

Задача 1: выбрать сведения о сотрудниках – фамилию, имя, адрес и номер телефона

- Для решения этой задачи нужно использовать запрос, который выбирает из таблицы **address** только те строки, в которых значения первичного ключа **address_id** совпадают со значениями внешнего ключа **address_id** в таблице **staff**. Тем самым решается задача выбора из таблицы **address** только адресов сотрудников:

```
SELECT first_name, last_name, address, phone  
FROM address, staff  
WHERE address.address_id = staff.address_id;
```

Связанные
(согласованные)
столбцы



	first_name	last_name	address	phone
▶	Mike	Hillyer	23 Workhaven Lane	14033335568
	Jon	Stephens	1411 Lillydale Drive	6172235589

Рис. 7. Результат работы запроса

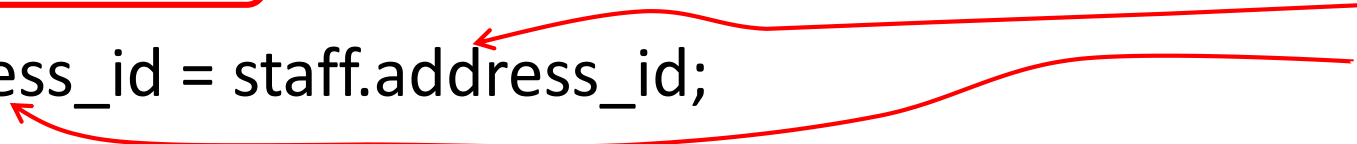
- Последующие версии стандартов SQL существенно расширили возможности соединения и добавили новые варианты предложения FROM. С использованием этих вариантов предыдущий пример может быть записан следующим образом.

```
SELECT first_name, last_name, address, phone
```

```
FROM address JOIN staff
```

```
ON address.address_id = staff.address_id;
```

Связанные
(согласованные)
столбцы



Результат работы данного запроса будет такой же, как на рис. 7.

- Вместо списка с разделяющими элементы запятыми, предложение FROM в приведенном примере применяет для описания операции соединения ключевое слово JOIN. Связанные столбцы, использующиеся при соединении, указаны в предложении ON, которое следует после предложения FROM.

Запросы с использованием отношения «предок-потомок»

4

- Среди многотабличных запросов наиболее распространены запросы к двум таблицам, связанным с помощью естественного отношения «предок-потомок».
- В реляционной базе данных отношение «предок-потомок» создают первичные и внешние ключи. Таблица, содержащая внешний ключ, является потомком, а таблица с первичным ключом – предком.
- Таким образом, если в условии отбора в запросе значение первичного ключа сравнивается со значением внешнего ключа, то говорят, что в запросе используется отношение «предок-потомок».
- Запрос, извлекающий сведения о сотрудниках (задача 1) является примером такого запроса.

Соединения с разными условиями отбора строк

5

- В многотабличном запросе можно комбинировать условие отбора, в котором задаются связанные столбцы, с другими условиями отбора.

Задача 2: выбрать сведения о сотрудниках – фамилию, имя, адрес и номер телефона, но включить в запрос только тех сотрудников, для которых идентификатор города `city_id` меньше 400 (см. рис. 3).

```
SELECT first_name, last_name, address, phone  
FROM address, staff  
WHERE address.address_id = staff.address_id AND city_id < 400;
```

	first_name	last_name	address	phone
▶	Mike	Hillyer	23 Workhaven Lane	14033335568

Рис. 8. Результат работы запроса

В предыдущем запросе условие соединения и условие отбора находятся в предложении WHERE.

При использовании другого синтаксиса условия соединения находятся в предложении ON, а условия отбора – в предложении WHERE, что делает запрос более легким для понимания:

```
SELECT first_name, last_name, address, phone  
FROM address JOIN staff  
ON address.address_id = staff.address_id  
WHERE city_id < 400;
```

Результат запроса будет таким же, как на рис. 8.

Естественные соединения

6

- На практике создатели баз данных часто используют одно и то же имя для столбца, который содержит идентификатор клиента или номер служащего, во всех таблицах, содержащих эти данные.
- В таком случае большинство соединений между двумя таблицами в такой базе данных являются соединениями по равенству на основе имен столбцов, имеющих в обеих таблицах. Такое соединение в стандарте SQL называется **естественным соединением (NATURAL JOIN)**.
- В задачах 1 и 2 в условии соединения – столбцы с одинаковыми названиями (`address_id`), поэтому для того, чтобы СУБД могла их различить при выполнении запроса, к ним добавлены названия таблиц, в которых они находятся.
- При использовании естественного соединения запрос из задачи 1 (см. слайд 12) будет выглядеть следующим образом:

```
SELECT first_name, last_name, address, phone  
FROM address NATURAL JOIN staff;
```

Результат запроса будет таким же, как на рис. 7.

Замечание 1. Натуральное соединение автоматически использует все столбцы с совпадающими именами. Поэтому при соединении двух таблиц необходимо убедиться, что в них присутствует только одна пара столбцов с одинаковым именем. В противном случае результат работы запроса с применением NATURAL JOIN может быть непредсказуемым.

Замечание 2. При внимательном изучении таблиц учебной базы Sakila, участвующих в запросе (address и staff), можно увидеть, что в них имеется еще одна пара одинаковых столбцов с именем last_update. Для того, чтобы выше приведенный запрос действительно работал, необходимо переименовать столбец last_update в одной из таблиц (address и staff). В противном случае запрос будет выполняться, но выдавать пустое множество строк.

Запросы к трем и более таблицам

7

Задача 3: выбрать сведения о сотрудниках – фамилию, имя, адрес, название города, номер телефона.

Данная задача отличается от **задачи 1** только необходимостью добавить в результат запроса название города, которое хранится в таблице **city**. Т. о. запрос для решения **задачи 3** будет выбирать данные из трех таблиц: **address, staff, city**.

```
SELECT first_name, last_name, address, city, phone
```

```
FROM address, staff, city
```

```
WHERE address.address_id = staff.address_id AND address.city_id = city.city_id;
```

	first_name	last_name	address	city	phone
▶	Mike	Hillyer	23 Workhaven Lane	Lethbridge	14033335568
	Jon	Stephens	1411 Lillydale Drive	Woodridge	6172235589

Рис. 9. Результат работы запроса

Альтернативная запись этого запроса более явно указывает каждое соединение и связанные столбцы:

```
SELECT first_name, last_name, address, city, phone
```

```
FROM address JOIN (staff, city)
```

```
ON (address.address_id = staff.address_id AND address.city_id = city.city_id);
```

Результат запроса будет таким же, как на рис. 9.

Преимуществом такого синтаксиса соединения в стандарте SQL является возможность определения порядка выполнения соединений.

Порядок же соединений в многотабличных запросах, записанных с помощью синтаксиса на слайде 23, значения не имеет. СУБД может соединить таблицу **address** с таблицей **staff**, а затем полученный результат с таблицей **city**. Или же она может сначала соединить таблицу **staff** с таблицей **city**, а затем результат – с таблицей **address**. В любом случае окончательный результат будет один и тот же, поэтому СУБД выполняет соединения в том порядке, который наиболее эффективен.

Прочие соединения по равенству

8

- Большое количество многотабличных запросов основано на отношениях «предок-потомок», но в SQL не требуется, чтобы связанные столбцы обязательно представляли собой пару «внешний ключ-первичный ключ».
- Любые два столбца из двух таблиц могут быть связанными, если только они имеют сравнимые типы данных.

Задача 4: выбрать адрес, индекс, название города и района (таблицы **address** и **city**) для тех адресов, у которых название района совпадает с названием города.

```
SELECT address, postal_code, city, district
FROM city, address
WHERE district = city;
```

Результатами запроса являются пары строк из таблиц **address** и **city**, имеющие одинаковые значения в столбцах **district** и **city** соответственно. Эти столбцы не являются ни внешними, ни первичными ключами. Тем не менее СУБД соединяет таблицы в соответствии с запросом.

	address	postal_code	city	district
▶	53 Idfu Parkway	42399	Nantou	Nantou
	478 Joliet Way	77948	Hamilton	Hamilton
	613 Korolev Drive	45844	Masqat	Masqat
	1531 Sal Drive	53628	Esfahan	Esfahan
	270 Amroha Parkway	29610	Osmaniye	Osmaniye
	320 Brest Avenue	43331	Kaduna	Kaduna
	262 A Corua (La C...	34418	Dhaka	Dhaka
	1780 Hino Boulevard	7716	Liepaja	Liepaja

Рис. 10. Результат работы запроса

Связанные столбцы, подобные приведенным в задаче 4, создают между двумя таблицами отношение «**многие-ко-многим**»: может быть много строк в таблице **address** с одинаковыми названиями районов, которым соответствует одно название города в таблице **city**, в то же время может быть много строк в таблице **city** с одинаковыми названиями городов, которым соответствует одно название района в таблице **address**.

Можно подвести следующие итоги.

- Соединение, созданное путем связи первичного ключа с внешним, всегда создает отношение предок-потомок «**один-ко-многим**».
- В прочих соединениях также могут существовать отношения «**один-ко-многим**», если по крайней мере в одной таблице связанный столбец содержит уникальные значения во всех строках.
- В общем случае в соединениях, созданных на основе произвольных связанных столбцов, существуют отношения «**многие-ко-многим**».

Соединение по неравенству

9

- Термин «соединение» применяется к любому запросу, который соединяет данные из двух таблиц базы данных путем сравнения значений в двух столбцах этих таблиц.
- Самыми распространенными являются соединения, созданные на основе равенства связанных столбцов (соединения по равенству).
- Но SQL позволяет соединять таблицы и с помощью других операций сравнения: $>$, $<$, $<>$ и др.

Задача 5: выбрать сведения о сотрудниках – фамилию, имя, адрес и номер телефона (аналогично задаче 1), у которых идентификатор адреса в таблице **staff** больше, чем идентификатор адреса в таблице **address**. Для упрощения понимания работы запроса в его результат добавлены идентификаторы адреса из двух таблиц.

```
SELECT staff.address_id, first_name, last_name, address.address_id,  
address, phone
```

```
FROM address, staff
```

```
WHERE staff.address_id > address.address_id;
```

	address_id	first_name	last_name	address_id	address	phone
▶	3	Mike	Hillyer	1	47 MySakila Drive	
	3	Mike	Hillyer	2	28 MySQL Boulevard	
	4	Jon	Stephens	1	47 MySakila Drive	
	4	Jon	Stephens	2	28 MySQL Boulevard	
	4	Jon	Stephens	3	23 Workhaven Lane	14033335568

Рис. 11. Результат работы запроса

Другой синтаксис записи запроса для решения задачи 5:

```
SELECT staff.address_id, first_name, last_name,  
address.address_id, address, phone  
FROM address JOIN staff  
ON (staff.address_id > address.address_id);
```

Результат запроса будет таким же, как на рис. 11.

Особенности многотабличных запросов

10

Многотабличные запросы, рассмотренные до сих пор, не требовали применения специальных синтаксических форм или каких-либо других особенностей языка SQL

Однако некоторые многотабличные запросы нельзя создать без использования дополнительных возможностей языка SQL.

- Иногда в многотабличных запросах требуется исключить неоднозначности при ссылках на столбцы, для этого используются **квалифицированные имена столбцов**
- В многотабличных запросах выбор всех столбцов **SELECT *** имеет особый смысл
- Для создания многотабличных запросов, связывающих таблицу саму с собой, можно воспользоваться **самосоединениями**
- В предложении FROM можно воспользоваться **псевдонимами таблиц**, чтобы упростить полные имена столбцов и обеспечить однозначность ссылок на столбцы в самосоединении

Квалифицированные имена столбцов

Как уже было отмечено выше, часто разные таблицы в базе данных содержат столбцы с одинаковыми именами.

Обычно при выполнении однотабличных запросов с такими столбцами затруднений не возникает, поскольку в предложении FROM можно явно указать, какая таблица имеется в виду.

Задача 6: выбрать из таблицы address идентификатор адреса, адрес и название района, где район начинается на букву «R».

```
SELECT address_id, address, district  
FROM address  
WHERE district LIKE 'R%';
```

	address_id	address	district
▶	26	28 Charlotte Amali...	Rabat-Sal-Zammo...
	64	81 Hodeida Way	Rajasthan
	124	241 Mosul Lane	Risaralda
	174	1877 Ezhou Lane	Rajasthan
	184	1410 Benin City Pa...	Risaralda
	195	1077 San Felipe de...	Rostov-na-Donu
	213	43 Dadu Avenue	Rajasthan
	228	60 Poos de Caldas ...	Rajasthan

Рис. 12. Результат работы запроса

Ситуация, где возникают проблемы с одинаковыми именами столбцов, возникает при решении задачи 1, если в предложении WHERE не указывать названия таблиц:

```
SELECT first_name, last_name, address, phone  
FROM address, staff  
WHERE address_id = address_id;
```

Такой запрос SQL является **неоднозначным**, при его выполнении (в СУБД MySQL) будет получено сообщение об ошибке: «Error Code: 1052. Column 'address_id' in where clause is ambiguous» («Код ошибки: 1052. Неоднозначный столбец address_id в выражении where»).

Чтобы исключить разночтения, при указании столбцов необходимо использовать их **квалифицированные (полные) имена**.

Квалифицированное имя столбца – имя, содержащее имя столбца и имя таблицы, в которой он находится

address.address_id
city.last_update

Исправленный вариант предыдущего запроса:

```
SELECT first_name, last_name, address, phone  
FROM address, staff  
WHERE address.address_id = staff.address_id;
```

Выборка всех столбцов

- Инструкция **SELECT *** может использоваться для выборки всех столбцов таблицы, указанной в предложении FROM. В многотабличном запросе звездочка означает выбор всех столбцов из всех таблиц, указанных в предложении FROM.

- Изменим запрос из задачи 1 следующим образом:

```
SELECT *
```

```
FROM address, staff
```

```
WHERE address.address_id = staff.address_id;
```

Таблица результатов такого запроса состоит из 20 столбцов, что, очевидно, затрудняет восприятие информации и не несет никакого практического значения.

	address_id	address	address2	district	city_id	postal_code	phone	location	last_update1	staff_id	first_name	last_name	address_id	picture	email	store_id	active	username	password	last_update
▶	3	23 Workhaven Lane	NULL	Alberta	300		14033335568	BLOB	2014-09-25...	1	Mike	Hillyer	3	BLOB	Mike.Hillyer@sakila...	1	1	Mike	8cb2237d...	2006-02-15 03:57:16
	4	1411 Lillydale Drive	NULL	QLD	576		6172235589	BLOB	2014-09-25...	2	Jon	Stephens	4	NULL	Jon.Stephens@saki...	2	1	Jon	NULL	2006-02-15 03:57:16

Рис. 12. Результат работы запроса

Многие диалекты SQL трактуют * как особый вид универсального имени столбца, которое преобразуется в список всех столбцов. В этих диалектах * можно использовать вместе с именем таблицы вместо списка полных имен столбцов.

В следующем запросе имя **staff.*** означает список имен всех столбцов таблицы **staff**.

```
SELECT staff.*, address
FROM address, staff
WHERE address.address_id = staff.address_id;
```

Таблица результатов запроса будет состоять из 12 столбцов – все 10 столбцов таблицы **staff** и один столбец таблицы **address**, указанный явно. Такой тип «**квалифицированных имен всех столбцов**» введен в стандарте SQL2 и поддерживается многими СУБД.

	staff_id	first_name	last_name	address_id	picture	email	store_id	active	username	password	last_update	address
▶	1	Mike	Hillyer	3	BLOB	Mike.Hillyer@sakila...	1	1	Mike	8cb2237d...	2006-02-15 03:57:16	23 Workhaven Lane
	2	Jon	Stephens	4	NULL	Jon.Stephens@saki...	2	1	Jon	NULL	2006-02-15 03:57:16	1411 Lillydale Drive

Рис. 13. Результат работы запроса

Псевдонимы таблиц

Псевдоним (**alias**, 'ālēēs') таблицы можно использовать в любом запросе:

- если запрос касается таблицы другого пользователя;
- если имя таблицы очень длинное и употреблять его в квалифицированных именах столбцов утомительно.

В запросе на выборку располагается сразу за настоящим именем таблицы.

```
SELECT EMPS.NAME, MGRS.NAME  
FROM SALESREPS EMPS, SALESREPS MGRS  
WHERE EMPS.MANAGER = MGRS.EMPL_NUM;
```

Если в предложении FROM содержится псевдоним таблицы, то в квалифицированном имени столбца должен использоваться именно псевдоним, а не реальное имя таблицы.

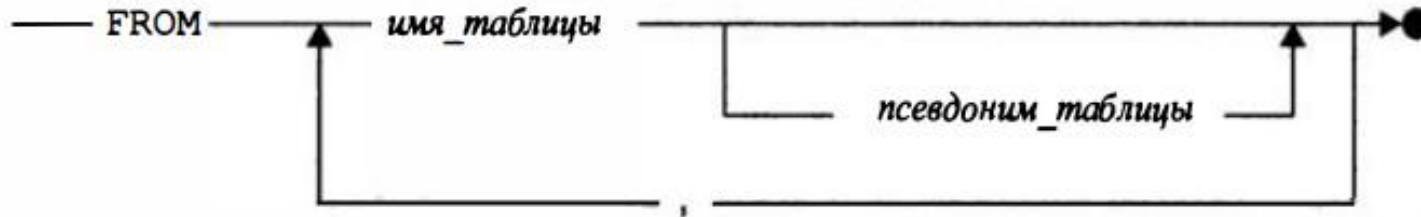


Рис. 14. Синтаксическая диаграмма предложения FROM для многотабличной инструкции SELECT, содержащей псевдонимы таблиц

1. В предложении FROM указываются все таблицы, из которых извлекаются данные. Любой столбец, указанный в инструкции SELECT, должен принадлежать одной из таблиц, заданных в предложении FROM. (Имеется исключение для внешних ссылок, содержащихся во вложенном запросе).
2. Предложение FROM содержит тег таблицы, который используется для указания таблицы в полном имени столбца в инструкции SELECT. Если в предложении FROM указывается псевдоним таблицы, то он становится ее тегом; в противном случае тегом становится имя таблицы в точности в том виде, в каком оно присутствует в предложении FROM.

- Единственное требование, предъявляемое к тегам таблиц в предложении FROM, состоит в том, что все они должны отличаться друг от друга.
- Стандарт SQL допускает необязательную вставку ключевого слова AS между именем и псевдонимом таблицы. В стандарте для того, что именуется псевдонимом таблицы (table alias), используется термин связанное имя (correlation name).
- Функция и смысл связанного имени те же, что и у псевдонимов; многие SQL-продукты используют термин псевдоним (alias), что лучше описывает суть решаемой псевдонимами таблиц задачи.
- Стандарт SQL определяет аналогичный метод создания альтернативных имен столбцов; в этом случае в стандарте псевдонимы столбцов именно так и называются – псевдонимами (alias).

Контрольные вопросы

1. Зачем нужны многотабличные запросы?
2. Что такое "простое соединение двух таблиц". Запишите запрос простого соединения двух таблиц двумя способами.
3. Что такое "отношение "предок-потомок"? Запишите запрос, использующий отношение "предок-потомок".
4. Что такое "соединение с условием отбора"? Запишите пример запроса.
5. Что такое "естественное соединение"? Запишите пример запроса.
6. Запишите пример запроса к трем таблицам.
7. Запишите пример запроса, реализующего соединение по неравенству.
8. Приведите пример квалифицированного имени столбца?
9. Запишите запрос, в котором используются псевдонимы столбцов и таблиц.

Список литературы

1. Грофф, Джеймс Р., Вайнберг, Пол Н., Оппель, Эндрю Дж. SQL : полное руководство, 3-е изд. : Пер. с англ. - М.: ООО «И.Д. Вильямс», 2015. - 960 с. : ил.
2. SELECT Statement. URL:
<https://dev.mysql.com/doc/refman/8.0/en/select.html>
3. Say NO to Venn Diagrams When Explaining JOINS. URL:
<https://blog.jooq.org/2016/07/05/say-no-to-venn-diagrams-when-explaining-joins/>