

КОНСПЕКТ ПО SQL




Типы баз данных:

1. Relational database – как информация организована
2. Transactional database – каким образом информация модифицируется

Основные составные элементы базы данных

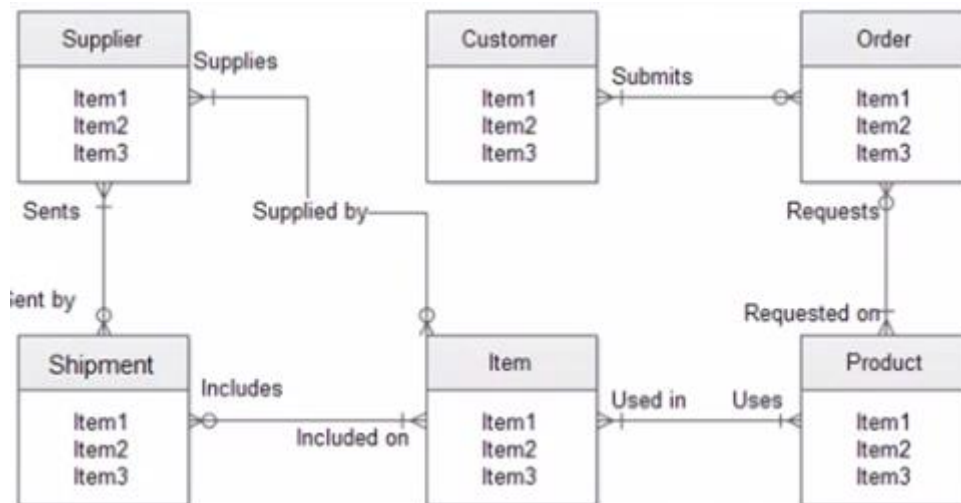
1. Сущность - таблица
2. Атрибут - признак
3. Связь – способ соединения таблиц: O-M, M-M, O-O

Relationships

- One-to-one 
- One-to-many 
- Many-to-many 

ER-диаграмма

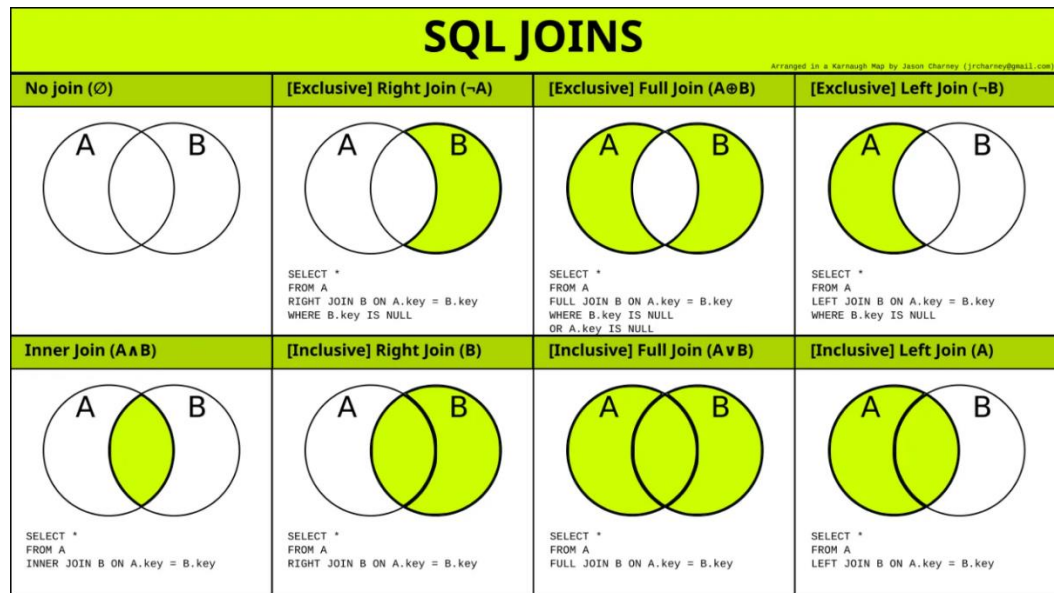
схема базы данных с таблицами, атрибутами, ключами, связями



1. Показывает отношения между таблицами
2. Бизнес процесс
3. Ключи :
 - a. Первичный ключ – колонка или колонки, которые однозначно идентифицируют запись
 - b. Вторичный ключ - колонка или колонки, которые идентифицируют запись не однозначно

Структура SQL-запроса:

SELECT -> FROM -> WHERE -> GROUP BY -> ORDER BY



```
SELECT *
FROM A
INNER JOIN B ON A.key = B.key
```

Операции со строкой:

Удаление пропусков (в начале и в конце, в конце, в начале): TRIM, RTRIM, LTRIM -

Конкатенация

Concatenations

Note:
SQL server supports + instead of ||

```
SELECT
  CompanyName
  ,ContactName
  ,CompanyName || ' (' || ContactName || ')'
FROM customers
```

	CompanyName	ContactName	CompanyName ' (' ContactName ')'
1	Alfreds Futterkiste	Maria Anders	Alfreds Futterkiste (Maria Anders)
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Ana Trujillo Emparedados y helados (Ana Trujillo)
3	Antonio Moreno Taquería	Antonio Moreno	Antonio Moreno Taquería (Antonio Moreno)
4	Around the Horn	Thomas Hardy	Around the Horn (Thomas Hardy)
5	Berglunds snabbköp	Christina Berglund	Berglunds snabbköp (Christina Berglund)
6	Blauer See Delikatessen	Hanna Moos	Blauer See Delikatessen (Hanna Moos)
7	Blondesddsl père et fils	Frédérique Citeaux	Blondesddsl père et fils (Frédérique Citeaux)
8	Bólido Comidas preparadas	Martín Sommer	Bólido Comidas preparadas (Martín Sommer)
9	Bon app'	Laurence Leblan	Bon app' (Laurence Leblan)
10	Bottom-Dollar Markets	Elizabeth Lincoln	Bottom-Dollar Markets (Elizabeth Lincoln)
11	B's Beverages	Victoria Ashworth	B's Beverages (Victoria Ashworth)
12	Cactus Comidas para llevar	Patricio Simpson	Cactus Comidas para llevar (Patricio Simpson)
13	Centro comercial Moctezuma	Francisco Chang	Centro comercial Moctezuma (Francisco Chang)
14	Chop-suey Chinese	Yang Wang	Chop-suey Chinese (Yang Wang)
15	Comércio Mineiro	Pedro Afonso	Comércio Mineiro (Pedro Afonso)

Операции с датой:

SQLite Date Time Functions

SQLite supports 5 date and time functions:

```
DATE(timestring, modifier, modifier, ...)  
TIME(timestring, modifier, modifier, ...)  
DATETIME(timestring, modifier, modifier, ...)  
JULIANDAY(timestring, modifier, modifier, ...)  
STRFTIME(format, timestring, modifier, modifier, ...)
```

Условный оператор (case)

Mimics if-then-else statement found
in most programming languages

Can be used in SELECT, INSERT,
UPDATE, and DELETE statements

```
CASE  
WHEN C1 THEN E1  
WHEN C2 THEN E2  
.  
.  
.  
ELSE [result else]  
END
```

```
CASE input_expression  
WHEN when_expression THEN result_expression [ ...n ]  
[ ELSE else_result_expression ]  
END
```

Overview of Views

A stored query

Can add or remove columns without changing schema

Use it to encapsulate queries

The view will be removed after database connection has ended

```
CREATE [TEMP] VIEW [IF NOT EXISTS]
view_name(column-name-list)
AS
select-statement;
```

Creating a View

```
CREATE VIEW my_view
AS
SELECT
r.regiondescription
,t.territorydescription
,e.Lastname
,e.Firstname
,e.Hiredate
,e.Reportsto
FROM Region r
INNER JOIN Territories t on r.regionid = t.regionid
INNER JOIN Employeeterritories et on t.TerritoryID = et.Te
```

```
SELECT *
FROM my_view
DROP VIEW my_view;
```

Wildcards

Wildcard Characters in SQL Server

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents a range of characters	c[a-b]t finds cat and cbt

All the wildcards can also be used in combinations!

Here are some examples showing different **LIKE** operators with '%' and '_' wildcards:

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a__%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

Продвинутые функции

- COALESCE (Oracle/PLSQL функция COALESCE возвращает первое ненулевое выражение из списка. Если все выражения определены как Null, то функция COALESCE вернет Null)
- MERGE

операция в языке T-SQL, при которой происходит обновление, вставка или удаление данных в таблице на основе результатов соединения с данными другой таблицы или SQL запроса. Другими словами, с помощью MERGE можно осуществить слияние двух таблиц, т.е. синхронизировать их.

Упрощённый синтаксис MERGE

```
MERGE <Основная таблица>
  USING <Таблица или запрос источника>
  ON <Условия объединения>
  [ WHEN MATCHED [ AND <Доп. условие> ]
    THEN <UPDATE или DELETE>
  [ WHEN NOT MATCHED [ AND <Доп. условие> ]
    THEN <INSERT> ]
  [ WHEN NOT MATCHED BY SOURCE [ AND <Доп. условие> ]
    THEN <UPDATE или DELETE> ] [ ...n ]
  [ OUTPUT ]
;
```

Важные моменты при использовании MERGE:

1. В конце инструкции MERGE обязательно должна идти точка с запятой (;) иначе возникнет ошибка;
2. Должно быть, по крайней мере, одно условие MATCHED;
3. Операцию MERGE можно использовать совместно с CTE (обобщенным табличным выражением);
4. В инструкции MERGE можно использовать ключевое слово OUTPUT, для того чтобы посмотреть какие изменения были внесены. Для идентификации операции здесь в OUTPUT можно использовать переменную \$action;
5. На все операции к основной таблице, которые предусмотрены в MERGE (удаления, вставки или обновления), действуют все ограничения, определенные для этой таблицы;
6. Функция @@ROWCOUNT, если ее использовать после инструкции MERGE, будет возвращать общее количество вставленных, обновленных и удаленных строк;
7. Для того чтобы использовать MERGE необходимо разрешение на INSERT, UPDATE или DELETE в основной таблице, и разрешение SELECT для таблицы источника;
8. При использовании MERGE необходимо учитывать, что все триггеры AFTER на INSERT, UPDATE или DELETE, определенные для целевой таблицы, будут запускаться.

- Оконные функции

Оконные функции начинаются с оператора OVER и настраиваются с помощью трёх других операторов: PARTITION BY, ORDER BY и ROWS. Про ORDER BY, PARTITION BY и его вспомогательные операторы LAG, LEAD, RANK мы расскажем подробнее.

Разница группировок и оконных функций на примере питона

Группировка:

```
df.groupby([поля_по_которым_группируем])[поля_для_которых_считаем_статистики].agg(словарь_с_агрегирующими_функциями)
```

Оконная функция (OVER, LAG, LEAD, RANK, row_number) :

```
df.groupby([поля_по_которым_группируем])[поля_для_которых_считаем_статистики].apply(агрегирующая_функция)
```

ПОСЛЕ GROUP BY РАЗМЕР ТАБЛИЦЫ МЕНЯЕТСЯ, ПОСЛЕ ОКОННОЙ ФУНКЦИИ – НЕТ

Пример:

```
1. SELECT
2.     Date
3.     , Medium
4.     , Conversions
5.     , SUM(Conversions) OVER(PARTITION BY Date ORDER BY Conversions ROWS BETWEEN CURRENT
      ROW AND 1 FOLLOWING) AS 'Sum'
6. FROM Orders
```

	Date	Medium	Conversions	Sum	
	10.05.2020	cpa	1	2	= 1+1
	10.05.2020	organic	1	3	= 1+2
	10.05.2020	cpc	2	2	= 2
	11.05.2020	cpa	1	2	= 1+1
	11.05.2020	direct	1	3	= 1+2
	11.05.2020	organic	2	5	= 2+3
	11.05.2020	cpc	3	3	= 3
	12.05.2020	cpc	1	3	= 1+2
	12.05.2020	organic	2	2	= 2