## Простой выбор данных

select \* from core.SourceData

Запрос звучит: “выбрать (select) всё (\*) из (from) таблицы (core.SourceData)

Можно выбрать не все поля, а только нужные нам. Для этого заменяем значок \*, на нужные названия столбцов. Они разделяются запятой, кроме последнего.

select

AtomId,

Date,

Clicks

from core.SourceData

## Выбор данных с условием

Условие записывается после основного запроса.

select \* from core.SourceData

where Date between '2016-10-01' and '2016-10-02'

Запрос звучит: “выбрать всё из таблицы ДанныеИзИсточников, где (where) дата между (between) значениями”.

Условия можно задавать по любому полю таблицы. Операторы могут быть разные:

* = — равно;
* <> — не равно;
* > — больше;
* < — меньше;
* between — диапазон между заданных значений;
* like — поле содержит какое-то значение;
* in — поле равно одному из значений.

select \* from core.Campaigns

where Name like N'%Интересы%'

Запрос звучит: “выбрать всё из талицы Кампании, где поле Имя содержит (like) символы ‘Интересы’”

Знаки % означают, что до и после запроса могут быть любые символы.

***(?)*** *Что будет, если убрать знак % в начале условия? — Будет означать, что нужно выбрать все записи, в которых поле Имя начинается с ‘Интересы’ и заканчивающиеся чем угодно.*

select \* from core.Campaigns

where Id in (5, 6, 7)

***(?)*** *Как звучит запрос? — Запрос звучит: “выбрать всё из талицы Кампании, где поле Идентификатор (Id) имеет значения (in) 5 или 6 или 7*

## Вставить данные

Иногда в таблицы нужно вставить новые строки.

Первый способ.

insert core.Campaign (Id, Name, OrderBy)

values (12, N'Не целевая', 980)

Запрос звучит: “вставить (insert) в таблицу Кампании, которая имеет столбцы (Идентификатор, Имя, Порядок) строку со значениями значения (values) (12, N'Не целевая', 980).

Символ N перед одинарными кавычками предупреждает базу, что в значении могут быть кириллические символы.

Второй способ. То же самое, но запись проще.

insert into core.Campaigns

values (12, N'Не целевая', 980)

Существуют автоматически создаваемые поля. Чаще всего это идентификаторы записей. Их не нужно указывать, не в перечислении столбцов, не в перечислении значений. Вы узнаете о том, что они автоматические, когда попробуйте добавить запись, всё будет вроде правильно, но программа выдаст ошибку.

## Обновить данные

Любые записи в таблице можно исправить — обновить.

update core.Atoms

set SegmentId = 321

where id = 123

Запрос звучит: “обнови (update) таблицу Атомы, запиши (set) в столбец ИдентификаторСегмента значение 321, где Идентификатор равен 123”.

## Соединить данные

Данные распределены по разным таблицам. Чтобы соединить их используем оператор join.

select \*

from core.Atoms as A

left outer join core.Sources as Src on A.SourceId = Src.Id

В данном примере мы присоединили к таблице Атомов таблицу Источников.

Запрос звучит: “выбрать все строки из таблицы Атомы, назовём её (as) А, присоединяем слева (left outer join) таблицу Источники, назовём её Scr, по полям ИдентификаторИсточника в таблице Атомов равен Идентификатору в таблице Источников и т.д.”

***(?)*** Присоедините к таблице атомов все необходимые справочники, чтобы можно было удобно смотреть статистику.

select \*

from core.Atoms as A

left outer join core.Sources as Src on A.SourceId = Src.Id

left outer join core.Campaigns as Camp on A.CampaignId = Camp.Id

left outer join core.Cities as C on A.CityId = C.Id

left outer join core.Segments as Seg on A.SegmentId = Seg.Id

***(?)*** Выберите только номера Атомов и названия городов, источников, кампаний и сегментов.

select

A.Id,

C.Name,

Src.Name,

Camp.Name,

Seg.Name

from core.Atoms as A

left outer join core.Sources as Src on A.SourceId = Src.Id

left outer join core.Campaigns as Camp on A.CampaignId = Camp.Id

left outer join core.Cities as C on A.CityId = C.Id

left outer join core.Segments as Seg on A.SegmentId = Seg.Id

## Удалить данные

Удаляются всегда строки с данными, не сама таблица. Саму таблицу тоже можно, но другим запросом, сейчас рассматривать не будем.

delete from core.Atoms

where id =1111

Запрос звучит: удалить (delete) строку из таблицы Атомов, где Идентификатор равен 1111

## Начать запрос и отменить действия

Иногда запрос нужно протестировать, для этого мы его отменяем после исполнения. Чтобы не заносить данные в базу «на живую».

В начало любого запроса вставляем begin tran, в конец rollback.

begin tran

delete from core.Atoms

where id = 1111

select \* from core.Atoms

rollback

Запрос звучит: “начни запрос (begin tran), удали строки из таблицы Атомы, где Идентификатор равен 1111, отмени запрос (rollback)”.

## Примеры запросов с джедайской встречи 03.10.2019

### Удаление

begin tran

delete from client.Leads where Date between '2016-08-01' and '2016-08-15'

select \* from client.Leads

rollback

### Переименовать что–то

BEGIN TRAN

UPDATE [core].[Segments]

SET Name = REPLACE(Name, N'Все факультеты',N'УНИК'),

GaDynamicSegment = REPLACE(Name, N'Все факультеты',N'УНИК')

GO

select \* from core.Segments

ROLLBACK

### Дубли

select  **from [client].[CrmPartners] where id in (select id from [client].[CrmPartners] group by id having count(**) > 1)

### Выборки со Взаимно:

На рассмотрении по дате сделки:

SELECT \*

FROM (

 SELECT C.\*, S.[StatusId], ROW\_NUMBER() OVER (PARTITION BY C.[Id] ORDER BY S.[Date] DESC) AS [RowNumber]

 FROM [crm].[Calls] AS C

 INNER JOIN [crm].[Comments] AS S ON S.[CallId] = C.[Id]

 WHERE C.[Date] >= N'2019-01-01' AND C.[Date] < N'2019-02-01'

 ) AS q

WHERE q.[RowNumber] = 1 AND q.[StatusId] NOT IN (5, 28, 29, 13)

----------------------------------------------------------------------

На рассмотрении по дате статуса:

SELECT \*

FROM (

 SELECT C.\*, S.[StatusId], S.[Date] AS [StatusDate], ROW\_NUMBER() OVER (PARTITION BY C.[Id] ORDER BY S.[Date] DESC) AS [RowNumber]

 FROM [crm].[Calls] AS C

 INNER JOIN [crm].[Comments] AS S ON S.[CallId] = C.[Id]

 ) AS q

WHERE q.[RowNumber] = 1 AND q.[StatusId] NOT IN (5, 28, 29, 13) AND q.[StatusDate] >= N'2019-01-01' AND q.[StatusDate] < N'2019-02-01'

----------------------------------------------------------------------

Сумма статусов по качественным:

SELECT q.[StatusId], COUNT(\*)

FROM (

 SELECT C.\*, S.[StatusId], S.[Date] AS [StatusDate], ROW\_NUMBER() OVER (PARTITION BY C.[Id] ORDER BY S.[Date] DESC) AS [RowNumber]

 FROM [crm].[Calls] AS C

 INNER JOIN [crm].[Comments] AS S ON S.[CallId] = C.[Id]

 WHERE C.[RejectId] IN (0, 2, 3)

 ) AS q

WHERE q.[RowNumber] = 1 AND q.[StatusId] NOT IN (5, 28, 29, 13) AND q.[StatusDate] >= N'2019-01-01' AND q.[StatusDate] < N'2019-02-01'

GROUP BY q.[StatusId]

----------------------------------------------------------------------

Сумма статусов всех сделок, созданных в январе:

SELECT q.[StatusId], COUNT(\*)

FROM (

 SELECT C.\*, S.[StatusId], S.[Date] AS [StatusDate], ROW\_NUMBER() OVER (PARTITION BY C.[Id] ORDER BY S.[Date] DESC) AS [RowNumber]

 FROM [crm].[Calls] AS C

 INNER JOIN [crm].[Comments] AS S ON S.[CallId] = C.[Id]

 WHERE C.[Date] >= N'2019-01-01' AND C.[Date] < N'2019-02-01'

 ) AS q

WHERE q.[RowNumber] = 1

GROUP BY q.[StatusId]

### Выборки с IBS

Отклики с известным телефоном и известным источником:

select R.Id, R.TelMob, R.CreateDate, R.IbsApplicantSourceId, R.IbsApplicantSourceName, C.Tel from client.CrmResumeDaxtra AS R

INNER JOIN [client].[Calls] AS C on R.TelMob = C.CallerTel

where R.AtomId = 0 and C.Tel in (N'+78002220337', N'+78002220447', N'+78002220449', N'+78002220453', N'+78002221529', N'+78002222834', N'+78002222859',

N'+78002223554', N'+78002223668', N'+78002224318', N'+78002224378', N'+78002224391', N'+78002224605', N'+78002224652', N'+78002224785', N'+78002225083', N'+78002225930',

N'+78002225942', N'+78002226258', N'+78002226453', N'+78002226708', N'+78002226754', N'+78002227339', N'+78002228902', N'+78005504986', N'+78005509482', N'+78005509632')

Order by R.CreateDate

Номера на которые звонили из кол–во звонков:

select DISTINCT(C.Tel), COUNT(C.CallerTel) from client.CrmResumeDaxtra AS R

INNER JOIN [client].[Calls] AS C on R.TelMob = C.CallerTel

where R.AtomId = 0

GROUP BY C.Tel

Проверка по конкретному номеру

select R.Id, R.AtomId, R.TelMob, R.CreateDate, C.DateTime, R.IbsApplicantSourceId, R.IbsApplicantSourceName, C.CallerTel , C.AtomId, C.Tel, A.Name

from client.CrmResumeDaxtra AS R

INNER JOIN (

SELECT \*, ROW\_NUMBER() OVER (PARTITION BY CallerTel ORDER BY [DateTime] DESC) AS [RowNumber]

 FROM [client].[Calls]

 WHERE CallerTel is not null and CallerTel!='' and Status = N'Ответили'

) AS C on R.TelMob = C.CallerTel

INNER JOIN [core].[Atoms] AS A on C.AtomId = A.Id

where R.AtomId = 0 and C.RowNumber = 1 and C.CallerTel in (N'79030831501') and C.RowNumber = 1

order by C.DateTime