*27. Какое системное представление позволяет получить сведения о SRID?*

Представление каталога - sys.spatial\_reference\_systems

*38. Как проходит обход иерархии в иерархических запросах в СУБД Oracle?*

*Порядок прохода записей – от корня обход в глубину*

***синтаксис***

*Select Column1, Column2, ...* *From <Table1>, <Table2>, ...* *Where <Condition3>*

*Connect By <Codition2>* *Start With <Condition1>*

1. *Сначала, команда получит все строки в таблице, соответствующие условиям в пункте* ***start with*** *(condition1), как корень дерева(корень или уровень 1).*
2. *После, для каждого корня требуется скан всей таблицы, чтобы получить следующие записи соответствующие условиям в пункте* ***connect by*** *(условие2) (****node*** *уровня 2), для каждого* ***node*** *уровня 2 сканируется все таблица, чтобы получить следующие записи соответствующие условиям в пункте* ***connect by*** *(****node*** *уровня 3), и так продолжается до тех пор, пока не останется больше записей соответстующих условиям* ***connect by*** *то предыдущий* ***node*** *будет уровнем листа в дереве.*
3. *Под конец, проверяем условия пункта* ***where*** *(уровень3), чтобы получить записи пункта "select tree".*

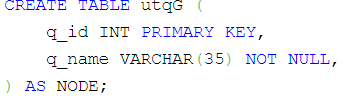
*48. Укажите виды и особенности создания таблиц в графовых базах данных в MS SQL Server.*

*ТАБЛИЦЫ УЗЛА*

*таблицы создаются аналогично обычным реляционным за исключением указания типа –* ***AS NODE*** *(т.е. узел).*

*При создании таблицы узлов, помимо пользовательских, автоматически создаются еще два псевдостолбца – graph\_id и $node\_id.*

*Столбец $node\_id является уникальным идентификатором узла, представленным в формате* ***JSON****.*

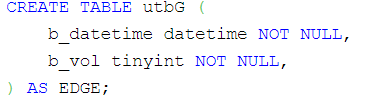
**

***ТАБЛИЦЫ РЕБРА***

*Она отличается от таблиц узлов типом – теперь это EDGE, а не NODE*

*к и для узлов, автоматически были созданы несколько псевдостолбцов, среди которых доступными пользователю являются:*

* ***$edge\_id*** *– идентификатор ребра, формируется автоматически;*
* ***$from\_id*** *– идентификатор узла, откуда исходит ребро;*
* ***$to\_id*** *– идентификатор узла, куда входит ребро.*

**

*49. Укажите особенности добавления ребер и узлов в графовых базах данных в MS SQL Server*

В узлы добавляются данные как обычно, например:.

insert into Person(id, name) values (1, ‘John’);  // 1 - id, John - имя

В рёбра записываются данные следующим образом:  
insert название\_таблицы\_рёбер ($to\_id, $from\_id)

values

((select $node\_id FROM Person where ID = 1), select select $node\_id FROM Person where ID = 2)); , где

$to\_id указывает направление ребра  
$from\_id указывает, от какого узла идёт ребро  
$node\_id - id узла

*50. Перечислите известные вам методы выборки в графовых базах данных в MS SQL Server.*

1. обычный select из таблицы узлов или рёбер (select \* from Person)
2. select с условием MATCH

-- Найти людей, которым нравятся рестораны, расположенные в том же городе, где они живут

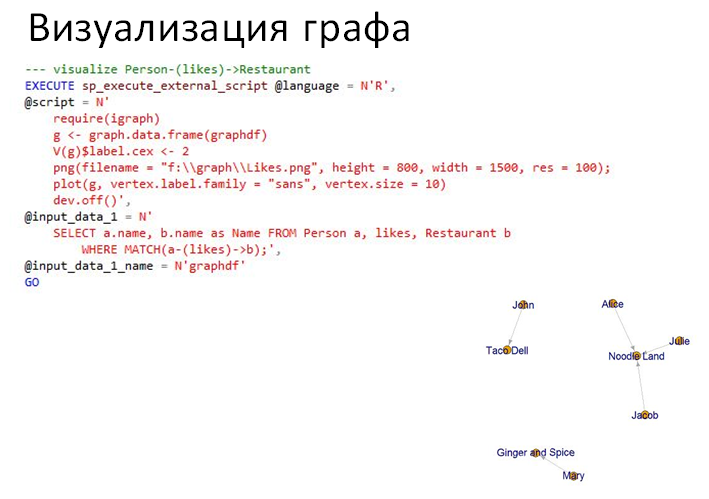
SELECT Person.name

FROM Person, likes, Restaurant, livesIn, City, locatedIn

WHERE MATCH (Person-(likes)->Restaurant-(locatedIn)->City AND Person-(livesIn)->City);

Здесь Person, Restaurant, Citi - узлы,

likes, locatedIn, livesIn - рёбра

1. с визуализацией **

*55. Как происходит преобразование XML данных в реляционные данные в MS SQL Server?*

Все узлы на 1 уровне становятся строками таблицы, все свойства внутри - значениями столбцов. Если свойство многоуровневое, то это считается внешним ключом на таблицу, которая тоже создаётся (это не точно).

*56. Как происходит преобразование реляционных данных в XML данные в MS SQL Server? Какие есть виды такого преобразования? Какие существуют директивы преобразования?*

Содержимое типа данных xml сериализуется обычным образом. То есть узлы элементов сопоставляются разметке элементов, а текстовые узлы сопоставляются текстовому содержимому.

Есть неявное преобразование  (из char, binary и тд), можно использовать CAST и CONVERT (явное преобразование).

Директивы: RAW, AUTO, EXPLICIT, PATH.

Режим RAW генерирует один элемент <row> для каждой строки в наборе строк, который возвращается оператором SELECT. Вы можете создать иерархию XML, написав вложенные запросы FOR XML.

Режим AUTO создает вложение в результирующем XML с помощью эвристики на основе способа, указанного в инструкции SELECT. Вы имеете минимальный контроль над формой сгенерированного XML.

Режим EXPLICIT позволяет лучше контролировать форму XML. Вы можете по желанию смешивать атрибуты и элементы при определении формы XML.

Режим PATH вместе с возможностью вложенного запроса FOR XML упрощает режим EXPLICIT. В режиме PATH имена столбцов или псевдонимы столбцов обрабатываются как выражения XPath. Эти выражения показывают, как значения отображаются в XML.