Администрирование баз данных и приложений

Объекты базы данных

Лекция 8

Основные объекты базы данных

- Пользователи и схемы
- Привилегии и роли
- Таблицы, столбцы, ограничения и типы данных (в том числе абстрактные типы данных)
- Последовательности
- Кластеры и хэш-кластеры
- Индексы
- Синонимы
- Представления
- Моментальные снимки и материализованные представления
- Связи баз данных
- Секции
- Процедуры, функции, пакеты
- Триггеры

Пользователи и схемы

- Учетная запись пользователя не является физической структурой
- Пользователям принадлежат объекты
- Схема – набор объектов, принадлежащий учетной записи пользователя
- Объекты создаются с правами учетных записей пользователей
- Учетные записи пользователей можно связать с учетными записями в ОС
- Двухкомпонентные имена – имя схемы.имя объекта

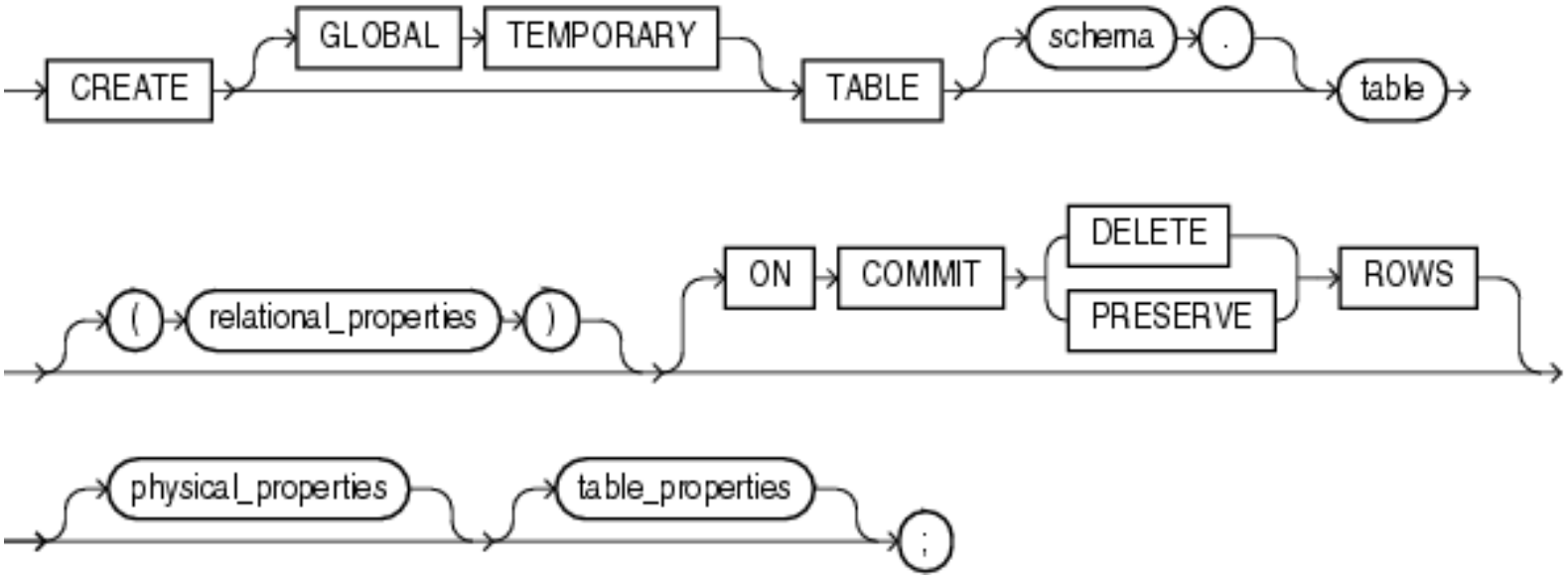
Таблицы

- Таблица – основная структура сохранения информации в БД
- Типы таблиц:
  - Традиционные таблицы (heap organized table)
  - Индекс-таблицы (index organized table)
  - Кластеризованные индекс-таблицы (index clustered table)
  - Кластеризованные хэш-таблицы (hash clustered table)
  - Отсортированные кластеризованные хэш-таблицы (sorted hash clustered table)
  - Вложенные таблицы (nested table)
  - Временные таблицы (temporary table)
  - Объектные таблицы
  - Внешние таблицы

Таблицы

- Может иметь до 1000 столбцов (<254)
- Может иметь неограниченное число строк
- Может иметь неограниченное число индексов
- Нет ограничения на число таблиц

Таблицы


CREATE
GLOBAL
TEMPORARY
TABLE
schema
.
table
(
relational_properties
)
ON
COMMIT
DELETE
PRESERVE
ROWS
physical_properties
table_properties
;

Параметры

```
CREATE TABLE Employee(
    emp_id char(12),
    emp_name nvarchar2(100),
    hire_date date,
    CONSTRAINT pk_emp_id primary key (emp_id)
)
organization index
nomonitoring
logging
pctthreshold 20
```

Параметры PCTFREE и PCTUSED

- Параметр PCTFREE – процент памяти блока, резервируемой для возможных обновлений строк, уже содержащихся в блоке
- Параметр PCTUSED – процент занятой части памяти блока

Управление параметрами заполнения блока

- Automatic segment space management - только PCTFREE
- Manual segment space management - PCTUSED, PCTTHRESHOLD, FREELISTS и др.

Таблицы

- Типы данных:
  - CHAR / NCHAR
  - VARCHAR2 / NVARCHAR2
  - DATE
  - INTERVAL DAY TO SECOND / INTERVAL YEAR TO MONTH TIMESTAMP
  - TIMESTAMP WITH TIME ZONE / TIMESTAMP WITH LOCAL TIME
  - NUMBER (A,B)
  - LONG RAW/ LONG / RAW
  - BLOB / CLOB / NCLOB
  - ROWID / UROWID

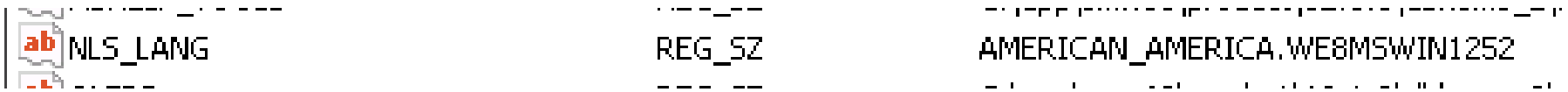
Типы данных ORACLE - символьные

CHAR	Символьное поле фиксированной длины до 2000 байт
NCHAR	Поле фиксированной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 2000 символов или 2000 байт в зависимости от набора символов.
VARCHAR2	Символьное поле переменной длины до 4000 байт
NVARCHAR2	Поле переменной длины для набора символов, состоящих из нескольких байт. Максимальный размер – 4000 символов или 4000 байт в зависимости от набора символов.

Поддержка национальных языков

- Переменная окружения NLS_LANG:
- NLS_LANG = language_territory.charset
  - Язык (LANGUAGE) – имена месяцев, имена дней, направление текста, сокращения для времени и дат. По умолчанию AMERICAN
  - Территория (TERRITORY) – настройки календаря, формат даты, формат денежной единицы. Если не указан, то будет взято значение, соответствующее языку (для RUSSIAN - CIS)
  - Набор символов (CHARACTER SET) – отображение символов, отображение и конвертация заглавных букв, порядок замещения символов при преобразовании – кодовая страница

Пример



- для Windows NLS-LANG может быть установлена в реестре и в переменных среды
- проверить, установлена ли переменная среды
- HOST ECHO %NLS_LANG%
- если ответ %NLS_LANG%, то не установлена
- переменная среды имеет приоритет перед установкой в реестре
- рекомендуется устанавливать только в реестре

Пример

```
SQL> SELECT * from NLS_INSTANCE_PARAMETERS;

PARAMETER
------------------------------
VALUE
------------------------------
NLS_LANGUAGE
AMERICAN

NLS_TERRITORY
AMERICA

NLS_SORT
```

	PARAMETER	VALUE
1	NLS_LANGUAGE	AMERICAN
2	NLS_TERRITORY	AMERICA
3	NLS_SORT	(null)
4	NLS_DATE_LANGUAGE	(null)
5	NLS_DATE_FORMAT	(null)
6	NLS_CURRENCY	(null)
7	NLS_NUMERIC_CHARACTERS	(null)
8	NLS_ISO_CURRENCY	(null)
9	NLS_CALENDAR	(null)
10	NLS_TIME_FORMAT	(null)
11	NLS_TIMESTAMP_FORMAT	(null)
12	NLS_TIME_TZ_FORMAT	(null)
13	NLS_TIMESTAMP_TZ_FORMAT	(null)
14	NLS_DUAL_CURRENCY	(null)
15	NLS_COMP	BINARY
16	NLS_LENGTH_SEMANTICS	BYTE
17	NLS_NCHAR_CONV_EXCP	FALSE

```
SQL> SELECT * from  NLS_DATABASE_PARAMETERS;

PARAMETER
                                                 VALUE
-------------------------------------------------- ----------------------------
----------------------------------
NLS_RDBMS_VERSION
                                                 12.1.0.1.0
NLS_NCHAR_CONV_EXCP
                                                 FALSE
NLS_LENGTH_SEMANTICS
                                                 BYTE
NLS_COMP
                                                 BINARY
NLS_DUAL_CURRENCY
                                                 $
NLS_TIMESTAMP_TZ_FORMAT
                                                 DD-MON-RR HH.MI.SSXFF AM TZR
NLS_TIME_TZ_FORMAT
                                                 HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_FORMAT
                                                 DD-MON-RR HH.MI.SSXFF AM
NLS_TIME_FORMAT
                                                 HH.MI.SSXFF AM
NLS_SORT
                                                 BINARY
NLS_DATE_LANGUAGE
                                                 AMERICAN
NLS_DATE_FORMAT
                                                 DD-MON-RR
NLS_CALENDAR
                                                 GREGORIAN
NLS_NUMERIC_CHARACTERS
                                                 .,
NLS_NCHAR_CHARACTERSET
                                                 AL16UTF16
NLS_CHARACTERSET
                                                 CL8MSWIN1251
NLS_ISO_CURRENCY
                                                 AMERICA
NLS_CURRENCY
                                                 $
NLS_TERRITORY
                                                 AMERICA
NLS_LANGUAGE
                                                 AMERICAN

20 rows selected.
```

	PARAMETER	VALUE
1	NLS_RDBMS_VERSION	12.1.0.1.0
2	NLS_NCHAR_CONV_EXCP	FALSE
3	NLS_LENGTH_SEMANTICS	BYTE
4	NLS_COMP	BINARY
5	NLS_DUAL_CURRENCY	$
6	NLS_TIMESTAMP_TZ_FORMAT	DD-MON-RR HH.MI.SSXFF AM TZR
7	NLS_TIME_TZ_FORMAT	HH.MI.SSXFF AM TZR
8	NLS_TIMESTAMP_FORMAT	DD-MON-RR HH.MI.SSXFF AM
9	NLS_TIME_FORMAT	HH.MI.SSXFF AM
10	NLS_SORT	BINARY
11	NLS_DATE_LANGUAGE	AMERICAN
12	NLS_DATE_FORMAT	DD-MON-RR
13	NLS_CALENDAR	GREGORIAN
14	NLS_NUMERIC_CHARACTERS	.,
15	NLS_NCHAR_CHARACTERSET	AL16UTF16
16	NLS_CHARACTERSET	CL8MSWIN1251
17	NLS_ISO_CURRENCY	AMERICA
18	NLS_CURRENCY	$
19	NLS_TERRITORY	AMERICA
20	NLS_LANGUAGE	AMERICAN

Пример

```
SQL>   SELECT * FROM NLS_SESSION_PARAMETERS;

PARAMETER                      VALUE
------------------------------ ----------------------------------------
---------------
NLS_LANGUAGE                   AMERICAN
NLS_TERRITORY                  AMERICA
NLS_CURRENCY                   $
NLS_ISO_CURRENCY               AMERICA
NLS_NUMERIC_CHARACTERS         .,
NLS_CALENDAR                   GREGORIAN
NLS_DATE_FORMAT                DD-MON-RR
NLS_DATE_LANGUAGE              AMERICAN
NLS_SORT                       BINARY
NLS_TIME_FORMAT                HH.MI.SSXFF AM
NLS_TIMESTAMP_FORMAT           DD-MON-RR HH.MI.SSXFF AM

PARAMETER                      VALUE
------------------------------ ----------------------------------------
---------------
NLS_TIME_TZ_FORMAT             HH.MI.SSXFF AM TZR
NLS_TIMESTAMP_TZ_FORMAT        DD-MON-RR HH.MI.SSXFF AM TZR
NLS_DUAL_CURRENCY              $
NLS_COMP                       BINARY
NLS_LENGTH_SEMANTICS           BYTE
NLS_NCHAR_CONV_EXCP            FALSE

17 rows selected.
```

	PARAMETER	VALUE
1	NLS_LANGUAGE	RUSSIAN
2	NLS_TERRITORY	RUSSIA
3	NLS_CURRENCY	р.
4	NLS_ISO_CURRENCY	RUSSIA
5	NLS_NUMERIC_CHARACTERS	,
6	NLS_CALENDAR	GREGORIAN
7	NLS_DATE_FORMAT	DD/MM/YY
8	NLS_DATE_LANGUAGE	RUSSIAN
9	NLS_SORT	RUSSIAN
10	NLS_TIME_FORMAT	HH24:MI:SSXFF
11	NLS_TIMESTAMP_FORMAT	DD.MM.RR HH24:MI:SSXFF
12	NLS_TIME_TZ_FORMAT	HH24:MI:SSXFF TZR
13	NLS_TIMESTAMP_TZ_FORMAT	DD.MM.RR HH24:MI:SSXFF TZR
14	NLS_DUAL_CURRENCY	р.
15	NLS_COMP	BINARY
16	NLS_LENGTH_SEMANTICS	BYTE
17	NLS_NCHAR_CONV_EXCP	FALSE

Пример

```
create table test (c char);
      insert into test (c) values ('A');
      commit;
      select c, ascii(c), asciistr(c) from test;
```

	C	ASCII(C)	ASCIISTR(C)
1	c	99	c
2	к	234	\043A
3	а	224	\0430
4	А	192	\0410

```
SQL> insert into test (c) values ('A');

1 row created.

SQL> insert into test (c) values ('a');

1 row created.

SQL> insert into test (c) values ('к');

1 row created.

SQL> commit;
```

	C	ASCII(C)	ASCIISTR(C)
1	c	99	c
2	к	234	\043A
3	а	224	\0430
4	А	192	\0410
5	€	136	\20AC
6		160	\00A0
7	?	63	?

```
SQL> select * from test;

C
-
c
к
к
A
a
?

7 rows selected.
```

NLS

- NLS_DATABASE_PARAMETERS – устанавливается при создании БД
- NLS_INSTANCE_PARAMETERS – устанавливается для экземпляра
- NLS_SESSION_PARAMETERS – устанавливается для сессии

```
-- https://docs.oracle.com/cd/A83908_02/NT816EE/DOC/nt.816/a73010/apb.htm
-- NLS_LANG = LANGUAGE_TERRITORY.CHARACTER_SET
-- AMERICAN_AMERICA.WE8MSWIN1252
```

128	Ђ	144	ђ	160		176	°	192	А	208	Р	224	а	240	р
129	Ѓ	145	‘	161	Ў	177	±	193	Б	209	С	225	б	241	с
130	‚	146	’	162	ў	178	І	194	В	210	Т	226	в	242	т
131	ѓ	147	“	163	Ј	179	і	195	Г	211	У	227	г	243	у
132	„	148	”	164	¤	180	ґ	196	Д	212	Ф	228	д	244	ф
133	…	149	•	165	Ґ	181	µ	197	Е	213	Х	229	е	245	х
134	†	150	–	166	¦	182	¶	198	Ж	214	Ц	230	ж	246	ц
135	‡	151	—	167	§	183	·	199	З	215	Ч	231	з	247	ч
136		152		168	Ё	184	ё	200	И	216	Ш	232	и	248	ш
137	‰	153	™	169	©	185	№	201	Й	217	Щ	233	й	249	щ
138	Љ	154	љ	170	Є	186	є	202	К	218	Ъ	234	к	250	ъ
139	‹	155	›	171	«	187	»	203	Л	219	Ы	235	л	251	ы
140	Њ	156	њ	172	¬	188	ј	204	М	220	Ь	236	м	252	ь
141	Ќ	157	ќ	173	-	189	S	205	Н	221	Э	237	н	253	э
142	Ћ	158	ћ	174	®	190	s	206	О	222	Ю	238	о	254	ю
143	Џ	159	џ	175	Ї	191	ї	207	П	223	Я	239	п	255	я

	C	ASCII(C)	ASCIISTR(C)
1	c	99	c
2	к	234	\043A
3	а	224	\0430
4	А	192	\0410
5	€	136	\20AC
6		160	\00A0
7	?	63	?

```
SQL> select * from test;

C
-
c
к
а
А
€

?

7 rows selected.
```

Пример

```
select c, ascii(c), asciistr(c), DUMP(c, 1010) from test;
```

	C	ASCII(C)	ASCIISTR(C)	DUMP(C,1010)
1	c	99	c	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 99
2	к	234	\043A	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 234
3	а	224	\0430	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 224
4	А	192	\0410	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 192
5	€	136	\20AC	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 136
6		160	\00A0	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 160
7	?	63	?	Typ=96 Len=1 CharacterSet=CL8MSWIN1251: 63

Типы данных ORACLE – символьные

LONG	Символьный, переменной длины, до 2GB, оставлен для совместимости
RAW(n)	Переменной длины, для бинарных данных n <= 2000 byte оставлен для совместимости
LONG RAW	Бинарные данные до 2GB
CLOB	Символьный тип большой объект до 4GB
NLOB	CLOB для многобайтных символов
BLOB	Большой двоичный объект до 4GB
BFILE	Указатель на двоичный файл операционной системы

Типы данных ORACLE – дата/время

DATE	7 байтовое поле фиксированной длины, используемое для хранения даты и времени
INTERVAL DAY TO SECOND	11 байтовое поле фиксированной длины для интервала времени: Дни, часы, минуты, секунды
INTERVAL YEAR TO MONTH TIMESTAMP	5 байтовое поле фиксированной длины для интервала времени: Годы и месяцы
TIMESTAMP WITH TIME ZONE	13 байтовое поле фиксированной длины Дата, время и настройки, связанные с часовым поясом.
TIMESTAMP WITH LOCAL TIME	7-11 байтовое поле переменной длины Дата и время, приведенные к часовому поясу базы данных

Типы данных ORACLE – числовые

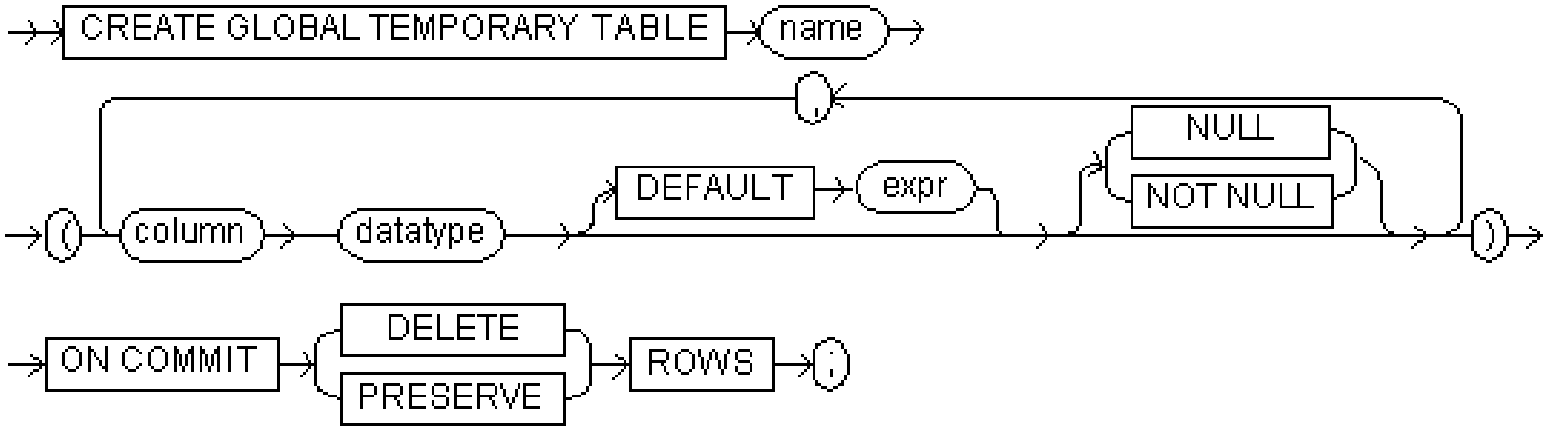
NUMBER(n, s)	Числовой тип переменной длины Точность n <= 38, общее количество цифр Масштаб s = [-84,127], количество цифр после запятой

Таблицы

- ROWID – 16-тиричный тип для уникального определения любой строки любой таблицы в БД
- Длина – 18 символов, которая делится на 4 элемента (6 + 3 + 6 + 3):
- OOOOOO – уникальный номер объекта в БД, которому принадлежит строка;
  - FFF– уникальный номер файла данных БД, где хранится строка;
  - BBBBBB –номер блока данных, который хранит строку, уникален на уровне файла данных БД;
  - RRR– адрес строки в блоке
  - UROWID – специальный 16-тиричный тип данных для адресации строк в таблицах, организованных по индексу

Временные таблицы

- Временные таблицы – механизм хранения данных в БД
- Состоит из столбцов и строк, как и обычная таблица



Временные таблицы

- Временные таблицы – глобальны
- Привилегии для создания временной таблицы CREATE TABLE
- Можно разместить временную таблицу в заданном табличном пространстве.
- Временные таблицы – это шаблон, хранящийся в словаре базы данных, для нее выделяется временный сегмент в (по умолчанию) TEMPORARY-табличном пространстве и для каждого пользователя свой.
- Каждый пользователь видит только свои данные (свой сегмент данных).

Временные таблицы

- Статичны: временные таблицы создаются (CREATE) один раз и существуют, пока их не удалят (DROP).
- DROP не получится, если таблица в этот момент используется другим пользователем.

Временные таблицы

- Временные таблицы бывают:
  - ON COMMIT PRESERVE ROWS – на время сеанса, данные существуют только на время сеанса, возможны все DML-операторы, TCL-операторы
  - ON COMMIT DELETE ROWS – на время транзакции, данные существуют только на время транзакции, возможны все DML-операторы, после выполнения COMMIT или ROLLBACK таблица становится пустой
- В начале сеанса временная таблица всегда пуста

Временные таблицы

- Для временных таблиц можно создавать триггеры
- Для временных таблиц можно указать констрейны (ограничения)
- Для временных таблиц можно создавать индексы

Временные таблицы

- Не могут быть индексно-организованными, нельзя секционировать, размещать в кластере.
- Данные повторного выполнения генерируются, но их количество пренебрежительно мало.

Временные таблицы

```
-- 15/01.sql

create global temporary  table gtemp_tlesson
(
 tlesson       char(10 byte) not null,
 tlesson_name varchar2(30 byte),
 constraint pk_temp_tlesson primary key (tlesson)
)

insert into gtemp_tlesson (tlesson, tlesson_name)
(
     select tlesson, tlesson_name
     from tlesson
)

select * from gtemp_tlesson


drop   table gtemp_tlesson;
```

Временные таблицы

```
-- 15/02.sql

create global temporary  table gtemp_tlesson as select * from tlesson;

insert into gtemp_tlesson (tlesson, tlesson_name)
(
      select tlesson, tlesson_name
      from tlesson
)
select * from gtemp_tlesson

drop   table gtemp_tlesson;
```

Временные таблицы

```
-- 15/03.sql
create global temporary  table gtemp_tlesson on commit preserve rows
      as select * from tlesson

insert into gtemp_tlesson  ( select * from tlesson)

select * from gtemp_tlesson
delete gtemp_tlesson
commit
drop   table gtemp_tlesson
```

```
Error starting at line 10 in command:
drop   table gtemp_tlesson
Error report:
SQL Error: ORA-14452: попытка создать, изменить или удалить индекс в уже используемой
14452. 00000 -  "attempt to create, alter or drop an index on temporary table already
*Cause:    An attempt was made to create, alter or drop an index on temporary
           table which is already in use.
*Action:   All the sessions using the session-specific temporary table have
           to truncate table and all the transactions using transaction
           specific temporary table have to end their transactions.
```

Временные таблицы

```
-- 15/04.sql
create global temporary  table gtemp_tlesson on commit delete rows
       as select * from tlesson

insert into gtemp_tlesson  ( select * from tlesson)

select * from gtemp_tlesson
delete gtemp_tlesson
commit
drop   table gtemp_tlesson
```

```
drop table gtemp_tlesson succeeded.
```

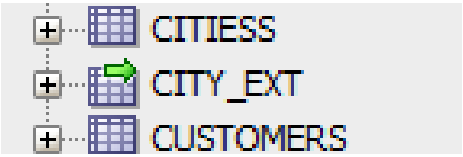
EXTERNAL TABLE

	A	B	C	D	E	F	G	H	I	J
1	Malisheve,Malisheve,42.4822,20.7458,Kosovo,XK,XKS,Malisheve,admin,,1901597212									
2	Prizren,Prizren,42.2139,20.7397,Kosovo,XK,XKS,Prizren,admin,,1901360309									
3	Zubin Potok,Zubin Potok,42.9144,20.6897,Kosovo,XK,XKS,Zubin Potok,admin,,1901608808									
4	Kamenice,Kamenice,42.5781,21.5803,Kosovo,XK,XKS,Kamenice,admin,,1901851592									
5	Viti,Viti,42.3214,21.3583,Kosovo,XK,XKS,Viti,admin,,1901328795									
6	Shterpce,Shterpce,42.2394,21.0272,Kosovo,XK,XKS,Shterpce,admin,,1901828239									
7	Shtime,Shtime,42.4331,21.0397,Kosovo,XK,XKS,Shtime,admin,,1901598505									
8	Vushtrri,Vushtrri,42.8231,20.9675,Kosovo,XK,XKS,Vushtrri,admin,,1901107642									
9	Dragash,Dragash,42.0265,20.6533,Kosovo,XK,XKS,Dragash,admin,,1901112530									
10	Podujeve,Podujeve,42.9111,21.1899,Kosovo,XK,XKS,Podujeve,admin,,1901550082									
11	Fushe Kosove,Fushe Kosove,42.6639,21.0961,Kosovo,XK,XKS,Fushe Kosove,admin,,1901134407									
12	Kacanik,Kacanik,42.2319,21.2594,Kosovo,XK,XKS,Kacanik,admin,,1901200321									
13	Kline,Kline,42.6217,20.5778,Kosovo,XK,XKS,Kline,admin,,1901230162									
14	Leposaviq,Leposaviq,43.1039,20.8028,Kosovo,XK,XKS,Leposaviq,admin,,1901974597									
15	Peje,Peje,42.6600,20.2922,Kosovo,XK,XKS,Peje,admin,,1901339694									
16	Rahovec,Rahovec,42.3994,20.6547,Kosovo,XK,XKS,Rahovec,admin,,1901336358									
17	Gjilan,Gjilan,42.4689,21.4633,Kosovo,XK,XKS,Gjilan,admin,,1901235642									
18	Lipjan,Lipjan,42.5217,21.1258,Kosovo,XK,XKS,Lipjan,admin,,1901682048									

EXTERNAL TABLE

```
---------------------- ORACLE_LOADER type
create or replace directory EXT_DIR as 'd:\external_dir';

CREATE TABLE city_ext (
  city          VARCHAR2(50),
  city_ascii    VARCHAR2(50),
  lat           VARCHAR2(50),
  lng           VARCHAR2(50),
  country       VARCHAR2(50),
  iso2          VARCHAR2(50),
  iso3          VARCHAR2(50),
  admin_name    VARCHAR2(50),
  capital       VARCHAR2(50),
  population    VARCHAR2(50),
  id            VARCHAR2(50)
)
ORGANIZATION EXTERNAL (
  TYPE ORACLE_LOADER
  DEFAULT DIRECTORY EXT_DIR
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    FIELDS TERMINATED BY ','
    MISSING FIELD VALUES ARE NULL)
  LOCATION ('data.csv')
)
PARALLEL 2  REJECT LIMIT UNLIMITED;
```



EXTERNAL TABLE

```
SELECT * FROM city_ext;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0,652 seconds

	CITY	CITY_ASCII	LAT	LNG	COUNTRY	ISO2	ISO3	ADMIN_NAME	CAPITAL	POPULATION	ID
1	Malisheve	Malisheve	42.4822	20.7458	Kosovo	XK	XKS	Malisheve	admin	(null)	1901597212
2	Prizren	Prizren	42.2139	20.7397	Kosovo	XK	XKS	Prizren	admin	(null)	1901360309
3	Zubin Potok	Zubin Potok	42.9144	20.6897	Kosovo	XK	XKS	Zubin Potok	admin	(null)	1901608808
4	Kamenice	Kamenice	42.5781	21.5803	Kosovo	XK	XKS	Kamenice	admin	(null)	1901851592
5	Viti	Viti	42.3214	21.3583	Kosovo	XK	XKS	Viti	admin	(null)	1901328795
6	Shterpce	Shterpce	42.2394	21.0272	Kosovo	XK	XKS	Shterpce	admin	(null)	1901828239
7	Shtime	Shtime	42.4331	21.0397	Kosovo	XK	XKS	Shtime	admin	(null)	1901598505
8	Vushtrri	Vushtrri	42.8231	20.9675	Kosovo	XK	XKS	Vushtrri	admin	(null)	1901107642
9	Dragash	Dragash	42.0265	20.6533	Kosovo	XK	XKS	Dragash	admin	(null)	1901112530
10	Podujeve	Podujeve	42.9111	21.1899	Kosovo	XK	XKS	Podujeve	admin	(null)	1901550082
11	Fushe Kosove	Fushe Kosove	42.6639	21.0961	Kosovo	XK	XKS	Fushe Kosove	admin	(null)	1901134407

EXTERNAL TABLE

```
UPDATE city_ext SET city = 'MINSK';
```

Script Output | Query Result

Task completed in 0,114 seconds

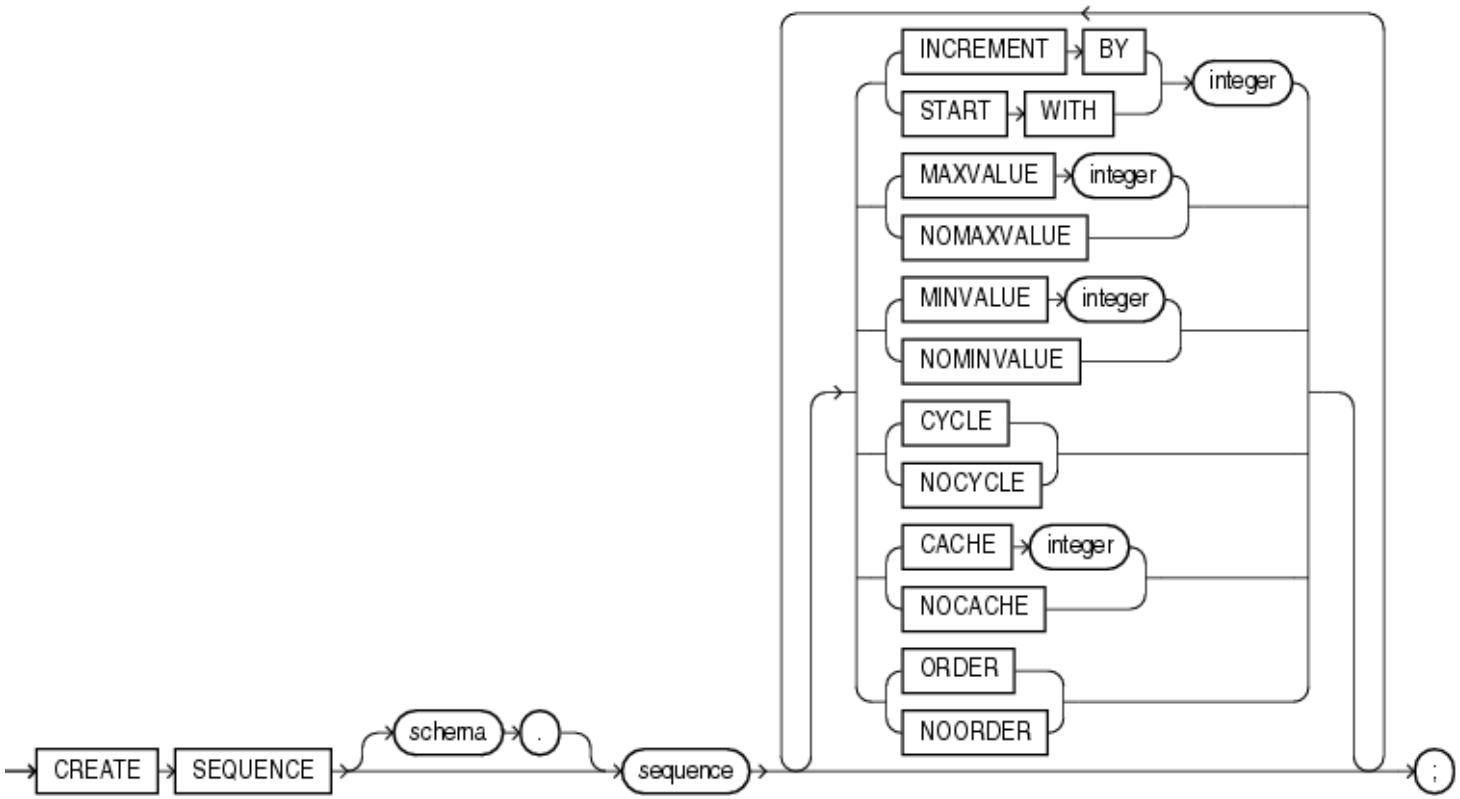
```
Directory EXT_DIR created.


Table CITY_EXT created.


Error starting at line : 31 in command -
UPDATE city_ext SET city = 'MINSK'
Error at Command Line : 31 Column : 8
Error report -
SQL Error: ORA-30657: операция во внешне-организованной таблице не поддерживается
30657.0000 -  "operation not supported on external organized table"
*Cause:    User attempted on operation on an external table which is
           not supported.
*Action:   Don't do that!
```

Последовательности

- Последовательность – объект базы данных, предназначенный для генерации числовой последовательности



Последовательности

- Привилегия CREATE SEQUENCE

```
SVVCORE@sh1> create sequence SVVCORE.SHEDULESSON_ID
  2    increment by     10
  3    start     with  100
  4    nomaxvalue          -- maxvalue
  5    nominvalue          -- minvalue
  6    nocycle             -- cycle
  7    cache 20            -- nocache
  8    order               -- noorder
  9  /

Последовательность создана.
```

Последовательности

```
SVVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;
select SHEDULESSON_ID.CURRVAL FROM DUAL
*
ошибка в строке 1:
ORA-08002: последов. SHEDULESSON_ID.CURRVAL еще не определен в этом сеансе


SVVCORE@sh1> select SHEDULESSON_ID.NEXTVAL from dual;

   NEXTVAL
----------
       100

SVVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;

   CURRVAL
----------
       100

SVVCORE@sh1> select SHEDULESSON_ID.NEXTVAL from dual;

   NEXTVAL
----------
       110

SVVCORE@sh1> select SHEDULESSON_ID.NEXTVAL, SHEDULESSON_ID.NEXTVAL from dual;

   NEXTVAL    NEXTVAL
---------- ----------
       120        120

SVVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;

   CURRVAL
----------
       120
```

Последовательности

- Представления словаря:
  - SYS.DBA_SEQUENCES
  - SYS.ALL_SEQUENCES
  - SYS.USER_SEQUENCES

Oracle Identity

```
CREATE  TABLE EMP  (
   ID NUMBER GENERATED BY DEFAULT ON NULL
     AS IDENTITY START WITH 1000
     MINVALUE 1000
     MAXVALUE 10000
     INCREMENT BY 10
     CACHE 20,
   NAME  VARCHAR2(100) NOT NULL,
   DEPT INT NOT NULL);
```

Oracle Identity

- GENERATED ALWAYS AS IDENTITY — значения определяются только сервером
- GENERATED BY DEFAULT AS IDENTITY — позволяет указать значение. Если значение не указано, то сервер назначит значение из последовательности
- GENERATED BY DEFAULT ON NULL AS IDENTITY — позволяет указать значение. Если значение в столбце не указано или явно указать значение NULL, то сервер назначит значение из последовательности.

Oracle Identity

```
INSERT INTO EMP(NAME, DEPT) VALUES('King', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(1020, 'Edwards', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(NULL, 'Dow', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(NULL, 'Jones', 20);
INSERT INTO EMP(ID, NAME, DEPT) VALUES(20, 'Forrest', 20);
INSERT INTO EMP(NAME, DEPT) VALUES('Lucas', 20);
COMMIT;
```

	ID	NAME	DEPT
1	1000	King	20
2	1020	Edwards	20
3	1010	Dow	20
4	1020	Jones	20
5	20	Forrest	20
6	1030	Lucas	20

Кластеры

- Таблицы, с которыми часто работают совместно, можно физически хранить совместно.
- Для этого создается кластер, который будет их содержать
- Строки из отдельных таблиц сохраняются в одних и тех же блоках, поэтому объединяющие запросы выполняются быстрее
- Уменьшается количество операций ввода-вывода
- Производительность операций вставки, обновления и удаления может быть ниже, чем для обычных таблиц
- Связанные столбцы называются кластерным ключом

Кластеры

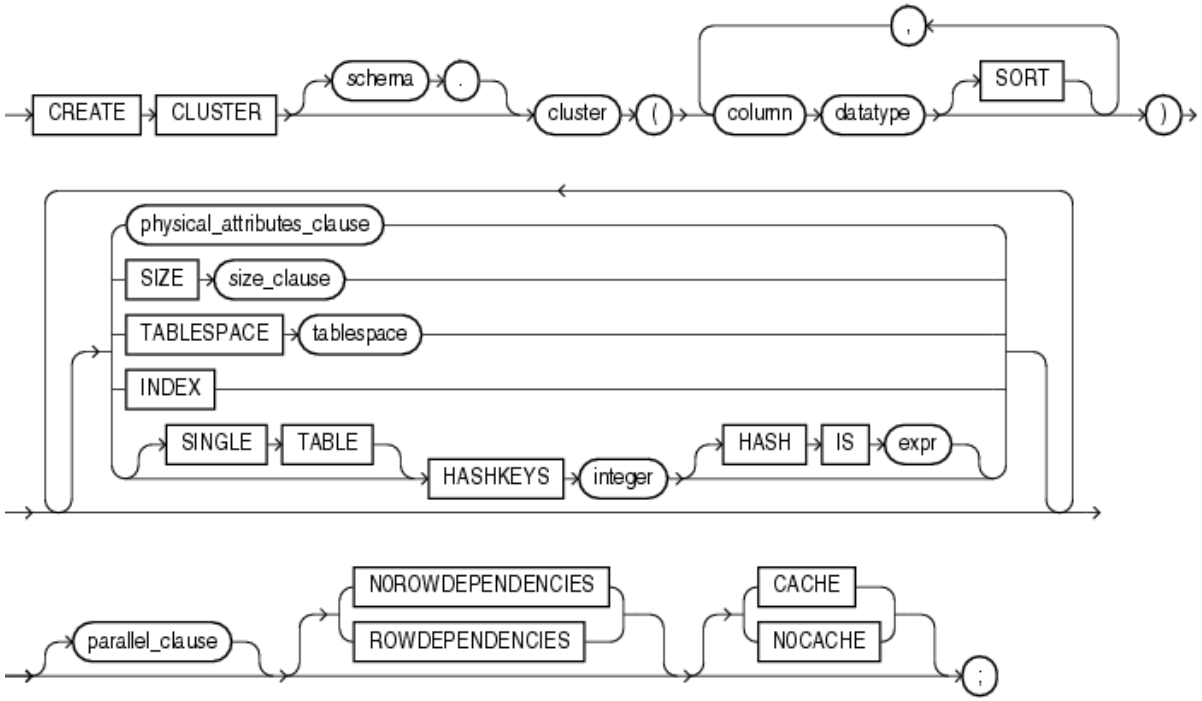
- Таблицы, с которыми часто работают совместно, можно физически хранить совместно.
- Для этого создается кластер, который будет их содержать
- Строки из отдельных таблиц сохраняются в одних и тех же блоках, поэтому объединяющие запросы выполняются быстрее
- Уменьшается количество операций ввода-вывода
- Производительность операций вставки, обновления и удаления может быть ниже, чем для обычных таблиц
- Связанные столбцы называются кластерным ключом

Хэш-кластеры

- Хэш-кластеры используют функции хэширования кластерного ключа строки для определения физической локализации места, где строку следует хранить
- Наибольшие преимущества – в запросах, использующих операции равенства:
- select Name from STUDENT where Id = 999;

Кластеры

- Кластер – объект БД, который хранит значения общих столбцов нескольких таблиц
- Создание CREATE CLUSTER
- Привилегия CREATE CLUSTER



Кластеры

```
SYSTEM@sh1> connect svvcore/svvcore@sh1;
Соединено.
sh1 - SVVCORE - 19.10.10
SVVCORE@sh1> create cluster svvcore.cluPULPIT
  2    (
  3     PULPIT CHAR(10)
  4    )
  5   hashkeys 100 ;

Кластер создан.
```

```
SVVCORE@sh1> CREATE TABLE SVVCORE.TEACHER
  2   (
  3    TEACHER       CHAR(10),
  4    TEACHER_NAME  VARCHAR2(50),
  5    PULPIT        CHAR(10)
  6   )
  7   CLUSTER SVVCORE.cluPULPIT(PULPIT);

Таблица создана.
```

```
SVVCORE@sh1> CREATE TABLE SVVCORE.PULPIT
  2  (
  3   PULPIT       CHAR(10),
  4   PULPIT_NAME  VARCHAR2(100),
  5   FACULTY      CHAR(10)
  6   )
  7   CLUSTER SVVCORE.cluPULPIT(PULPIT)
  8   COMPRESS
  9   --  CACHE не возможет с кластером
 10   ;

Таблица создана.
```

Кластеры

```
SVVCORE@shl> select CLUSTER_NAME,
  2         OWNER,
  3         TABLESPACE_NAME,
  4         CLUSTER_TYPE,
  5         CACHE
  6  from DBA_CLUSTERS;

CLUSTER_NAME                   OWNER                          TABLESPACE_NAME                CLUST CACHE
------------------------------ ------------------------------ ------------------------------ ----- -----
C_COBJ#                        SYS                            SYSTEM                         INDEX     N
C_TS#                          SYS                            SYSTEM                         INDEX     N
C_FILE#_BLOCK#                 SYS                            SYSTEM                         INDEX     N
C_USER#                        SYS                            SYSTEM                         INDEX     N
C_OBJ#                         SYS                            SYSTEM                         INDEX     N
C_MLOG#                        SYS                            SYSTEM                         INDEX     N
C_TOID_VERSION#                SYS                            SYSTEM                         INDEX     N
C_RG#                          SYS                            SYSTEM                         INDEX     N
C_OBJ#_INTCOL#                 SYS                            SYSTEM                         INDEX     N
SMON_SCN_TO_TIME               SYS                            SYSTEM                         INDEX     N
CLUPULPIT                      SVVCORE                        TSDATASVV                      HASH      N

11 строк выбрано.
```

Кластеры

```
SVVCORE@sh1> select tablespace_name,
  2         segment_type,
  3         count(*),
  4         sum(blocks),
  5         sum(extents),
  6         sum(bytes)
  7  from dba_segments
  8  where tablespace_name='TSDATASVV' OR  tablespace_name ='TSTEMPSVV'
  9  group by tablespace_name, segment_type
 10  order by tablespace_name, segment_type;

TABLESPACE_NAME                SEGMENT_TYPE         COUNT(*) SUM(BLOCKS) SUM(EXTENTS) SUM(BYTES)
------------------------------ ------------------ ---------- ----------- ------------ ----------
TSDATASVV                      CLUSTER                     1         104           13     851968
TSDATASVV                      INDEX                      10          80           10     655360
TSDATASVV                      TABLE                       8          64            8     524288
```

Синонимы

- Синоним – способ обращаться к объекту базы данных без указания обязательной полной идентификации объекта (хост – экземпляр – владелец – объект).
- Частный синоним принадлежит пользователю, который его создал.
- Публичный синоним используется совместно всеми пользователями базы данных.

Синонимы

- Привилегия – CREATE (PUBLIC) SYNONYM
- Создание – CREATE (PUBLIC) SYNONYM
- Допустимость синонима не проверяется сервером при создании!
- Представление словаря dba.synonyms

Синонимы

- Может указывать на:
  - Таблицы,
  - Процедуры,
  - Функции,
  - Последовательности,
  - Представления
  - Пакеты
  - Объекты в локальной или удаленной базе данных

Синонимы

```
GRANT CREATE SYNONYM TO  RLSVVCORE

CREATE SYNONYM T1  FOR  SVVCORE.TEACHER;

SELECT * FROM T1;

DROP  SYNONYM T1;


GRANT CREATE PUBLIC  SYNONYM,
      DROP   PUBLIC  SYNONYM     TO  RLSVVCORE


CREATE PUBLIC  SYNONYM T1  FOR  SVVCORE.TEACHER;
```

```
SQL> connect isuscore/isuscore@isus
Соединено.
SQL> select * from t1;
```

Представления

- Представление – хранимый запрос
- Можно обращаться, как к обычной таблице
- Данные хранятся в таблице
- Добавляют уровень защиты данных
- Скрывают сложность данных
- Скрывают имена столбцов таблиц

Представления

- Привилегия – CREATE VIEW
- Создание – CREATE (OR REPLACE) VIEW
- FORCE – создает представление, независимо от того, существуют ли таблицы и есть ли права
- NOFORCE – по умолчанию
- WITH CHECK OPTION – указывает, что будут вставлены или изменены строки, которые будут выбираться через это представление
- READ ONLY

Представления

```
CREATE VIEW VPULPIT_NAME
AS SELECT PUPIT_NAME FROM SVVCORE.PULPIT;
```

```
SQL Error: ORA-00942: таблица или представление пользователя не существует
00942. 00000 - "table or view does not exist"
```

```
CREATE FORCE VIEW VPULPIT_NAME
AS SELECT PUPIT_NAME FROM SVVCORE.PULPIT;
```

```
SQL Command: CREATE VIEW
Failed: Warning: выполнение завершено с предупреждением
```

```
SELECT * FROM VPULPIT_NAME;
```

```
ORA-04063: view "SVVCORE.VPULPIT_NAME" имеет ошибки
```

```
SELECT OWNER, VIEW_NAME, TEXT FROM DBA_VIEWS WHERE OWNER='SVVCORE'
```

SVVCORE	VPULPIT_NAME	SELECT PUPIT_NAME FROM SVVCORE.PULPIT
SVVCORE	VTEACHER_NAME	SELECT TEACHER_NAME FROM SVVCORE.TEACHER

Представления

```
CREATE TABLE SVVCORE.PULPIT
(
 PULPIT       CHAR(10) PRIMARY KEY,
 PULPIT_NAME  VARCHAR2(50)
)

SELECT * FROM VPULPIT_NAME;
```

ORA-04063: view "SVVCORE.VPULPIT_NAME" имеет ошибки

```
ALTER VIEW vpulpit_name COMPILE

Failed: Warning: выполнение завершено с предупреждением
 ALTER VIEW vpulpit_name succeeded.
```

```
CREATE OR REPLACE  VIEW   VPULPIT_NAME_FK
AS SELECT  PULPIT_NAME  FROM  SVVCORE.PULPIT WHERE PULPIT='ФК'
WITH CHECK OPTION CONSTRAINT PULPLIT_P;
```

Материализованные представления

- Механизм переписывания запросов (QUERY REWRITE) – подмена запроса перед выполнением
  - *QUERY REWRITE ENABLED*
  - *QUERY REWRITE INTEGRITY*
    - *ENFORCED*
    - *TRUSTED*
    - *STALE TOLERATED*

```
SELECT NAME, VALUE FROM V$PARAMETER
  WHERE NAME LIKE '%rewrite%';
```

	NAME	VALUE
1	query_rewrite_enabled	TRUE
2	query_rewrite_integrity	enforced

Материализованные представления

- Материализованное представление — физический объект базы данных, содержащий результат выполнения запроса
- Привилегия – CREATE MATERIALIZED VIEW
- Создание – CREATE MATERIALIZED VIEW
- BUILD IMMEDIATE – создает представление в момент выполнения оператора
- START WITH – показывает, когда выполнится в первый раз (если не был построен сразу)
- NEXT– показывает, когда выполнится в следующий раз
- Далее – в разницу времени между START WITH и NEXT

Материализованные представления

```
GRANT CREATE MATERIALIZED VIEW TO A1;
GRANT QUERY REWRITE TO A1;


 CREATE MATERIALIZED VIEW EMP_SUMMARY
   BUILD IMMEDIATE
   REFRESH COMPLETE
   ENABLE QUERY REWRITE
   AS SELECT COUNT(EMP_ID) FROM EMPLOYEE;
```

Материализованные представления

- REFRESH
  - COMPLETE — полное обновление данных из базовых таблиц
  - REFRESH FAST – используются журналы фиксации изменений базовых таблиц
  - REFRESH FORCE – попытка быстрого обновления; если быстрое обновление невозможно, то выполняется полное обновление

Материализованные представления

- Обновление явное
- Обновление неявное по расписанию
- REFRESH
  - ON COMMIT – обновление при COMMIT
  - ON DEMAND – обновление по требованию

```
ALTER  MATERIALIZED VIEW EMP_SUMMARY
  REFRESH COMPLETE ON DEMAND
  NEXT SYSDATE + NUMTODSINTERVAL (2, 'MINUTE');

EXECUTE DBMS_MVIEW.REFRESH ('EMP_SUMMARY');
```

Материализованные представления

- *REFRESH FAST*
- Создать журналы материализованного представления - вспомогательные таблицы, накапливающие сведения об изменениях, которые происходят в базовых таблицах
- Создать материализованное представление

```
CREATE MATERIALIZED VIEW LOG ON EMPLOYEE
  WITH ROWID, PRIMARY KEY
  INCLUDING NEW VALUES;

CREATE MATERIALIZED VIEW EMP_SALARY_LIST
  REFRESH FAST ON COMMIT
  ENABLE QUERY REWRITE
   AS SELECT EMP_ID, SALARY  FROM EMPLOYEE;
```

Материализованные представления

- Просмотр
  - *USER_MVIEWS*
  - *USER_MVIEW_LOGS*
- Удалить
  - *DROP MATERIALIZED VIEW*

Индексы

- Индекс – структура базы данных, используемая сервером для быстрого поиска строки в таблице

Оптимизация запросов

- Оптимизатор запроса — встроенное в СУБД программное обеспечение, которое определяет наиболее эффективный способ выполнения SQL-выражения
- План выполнения запроса — последовательность шагов или инструкций СУБД, необходимых для выполнения SQL-выражения
- Стоимость выполнения запроса — наилучшая оценка времени, необходимого для выполнения оператора, полученная оптимизатором

Оптимизация запросов

- Оптимизатор разрабатывает множество потенциальных планов выполнения SQL-запросов
- Для каждого потенциального плана выполнения запроса оптимизатор оценивает стоимость его выполнения

Оптимизация запросов

- Для оценки применяется информация о ресурсах:
  - дисковый ввод/вывод
  - загрузка центрального процессора
  - оперативная память
- Выбор оптимального плана выполнения запроса зависит от:
  - объема предполагаемого набора данных
  - расположение данных
  - структуры доступа к данным

Оптимизация запросов

- Оптимизатор определяет наилучший план выполнения разными способами:
  - полное сканирование таблицы
  - использование индексов
  - различные типы соединения

Оптимизация запросов

- Селективность (избирательность)
- Кардинальность

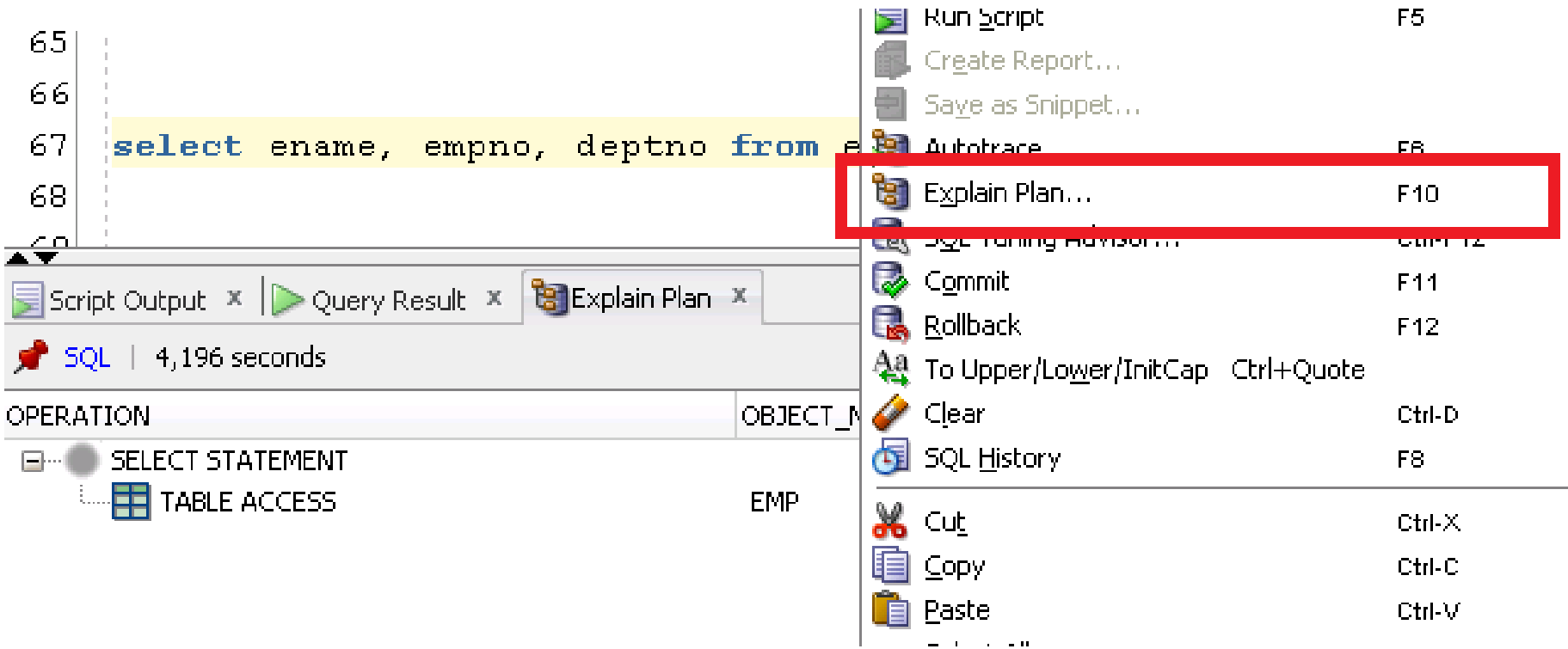
Селективность

- Селективность таблицы — значение, представляющее долю строк таблицы, удовлетворяющих определенному условию выбора
- Селективность таблицы связана с условием выбора строк
- Селективность индекса — значение, представляющее отношение количества уникальных значений индексируемых столбцов к общему числу строк таблицы
- Селективность индекса показывает долю строк от общего числа строк в таблице, которое приходится на одно значение индекса

Кардинальность

- Кардинальность — количество строк, возвращаемых после каждой операции плана выполнения запроса
- Значение кардинальности равно произведению селективности на количество обработанных строк

План запроса



План запроса

```
67  select ename, empno, deptno
68  from emp
69  where empno = 7839;
70
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
TABLE ACCESS	EMP	BY INDEX ROWID	1
INDEX	EMP_PK	UNIQUE SCAN	0
Access Predicates			
EMPNO=7839			

План запроса

```
67  select ename, empno, deptno
68  from emp
69  --where empno = 7839;
70
71
```

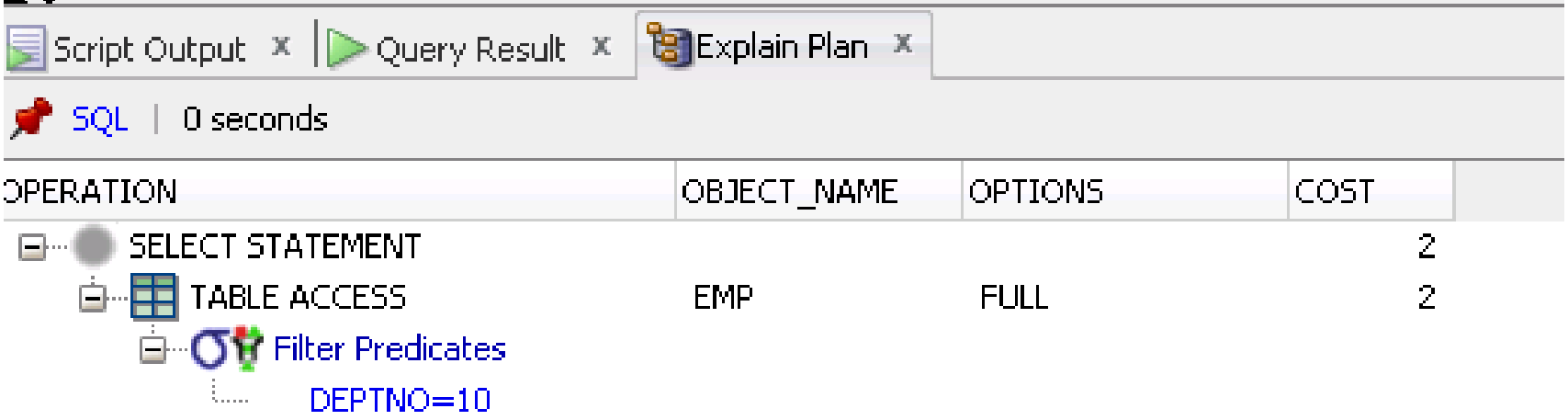
Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			2
TABLE ACCESS	EMP	FULL	2

План запроса

```
67  select ename, empno, deptno
68  from emp
69  where deptno = 10;
70
71
```



План запроса

```
304  select c.company, o.order_num, o.order_date, o.amount
305  from customers c join orders o on c.cust_num = o.cust
306  where o.amount <50;
307
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			3
HASH JOIN			3
Access Predicates			
C.CUST_NUM=O.CUST			
NESTED LOOPS			
NESTED LOOPS			3
STATISTICS COLLECTOR			
TABLE ACCESS	ORDERS	FULL	2
Filter Predicates			
O.AMOUNT<50			
INDEX	SYS_C0010526	UNIQUE SCAN	0
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1
TABLE ACCESS	CUSTOMERS	FULL	1

Индексы

- Типы индексов:
  - Табличный (B*Tree) индекс
  - Битовый индекс
  - Функциональный индекс

Индексы

- Табличный индекс (B*Tree) структурирован в виде сбалансированного дерева
- Листовой блок содержит индексированные значения столбца и соответствующий ему идентификатор строки (RowId)
- Предназначен для индексирования уникальных столбцов или столбцов с высокой селективностью

```
create index or_amount
on orders(amount);
```

Индексы

- Битовый индекс создает битовые карты для каждого возможного значения столбца, где каждому биту соответствует строка, а значение бита 1 (0) означает, что соответствующая строка содержит (не содержит) индексируемое значение
- Предназначен для индексирования столбцов с низкой селективностью
- Не подходит для таблиц с частым обновлением
- Хорошо подходят для хранилищ данных

Индексы

```
SVVCORE@sh1> create bitmap index ibmAUDITORIUM_TYPE on AUDITORIUM(AUDITORIUM_TYPE);

Индекс создан.

SVVCORE@sh1> create index idxTEACHER_NAME on TEACHER(TEACHER_NAME ASC);

Индекс создан.

SVVCORE@sh1> select index_name,
  2         index_type,
  3         table_name,
  4         uniqueness
  5  from DBA_INDEXES
  6  where owner = 'SVVCORE'
  7  /

INDEX_NAME                     INDEX_TYPE                  TABLE_NAME                     UNIQUENES
------------------------------ --------------------------- ------------------------------ ---------
PK_FACULTY                     IOT - TOP                   FACULTY                        UNIQUE
PK_PULPIT                      NORMAL                      PULPIT                         UNIQUE
IDXTEACHER_NAME                NORMAL                      TEACHER                        NONUNIQUE
PK_TEACHER                     NORMAL                      TEACHER                        UNIQUE
PK_AUDITORIUM                  NORMAL                      AUDITORIUM                     UNIQUE
IBMAUDITORIUM_TYPE             BITMAP                      AUDITORIUM                     NONUNIQUE
PK_AUDITORIUM_TYPE             NORMAL                      AUDITORIUM_TYPE                UNIQUE
PK_SUBJECT                     NORMAL                      SUBJECT                        UNIQUE

8 строк выбрано.
```

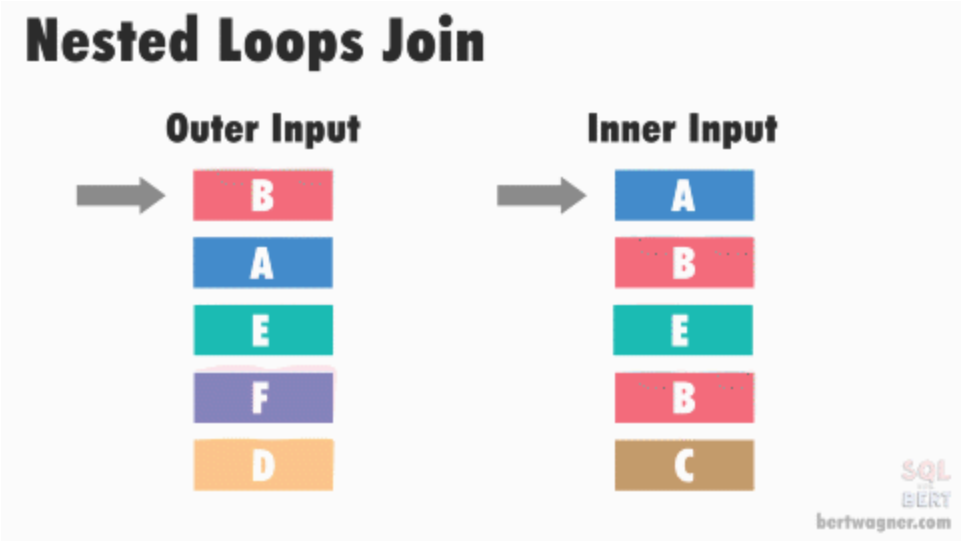
Индексы

- Функциональный индекс – предварительно вычисляют значения функции по заданному столбцу и сохраняют результат в индексе
- LOWER(NAME) / UPPER (NAME)

Индексы

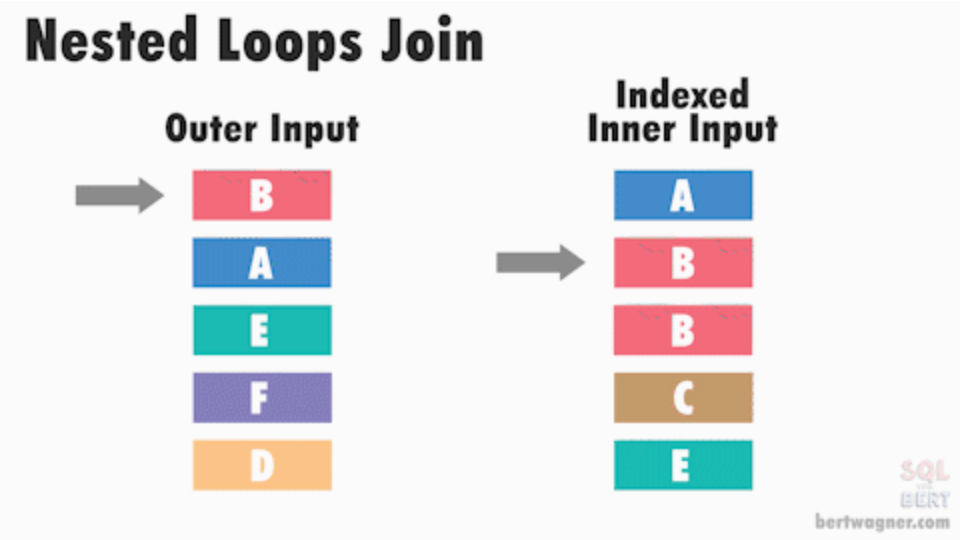
- Функциональный индекс – предварительно вычисляют значения функции по заданному столбцу и сохраняют результат в индексе
- LOWER(NAME) / UPPER (NAME)

Nested Loops Join

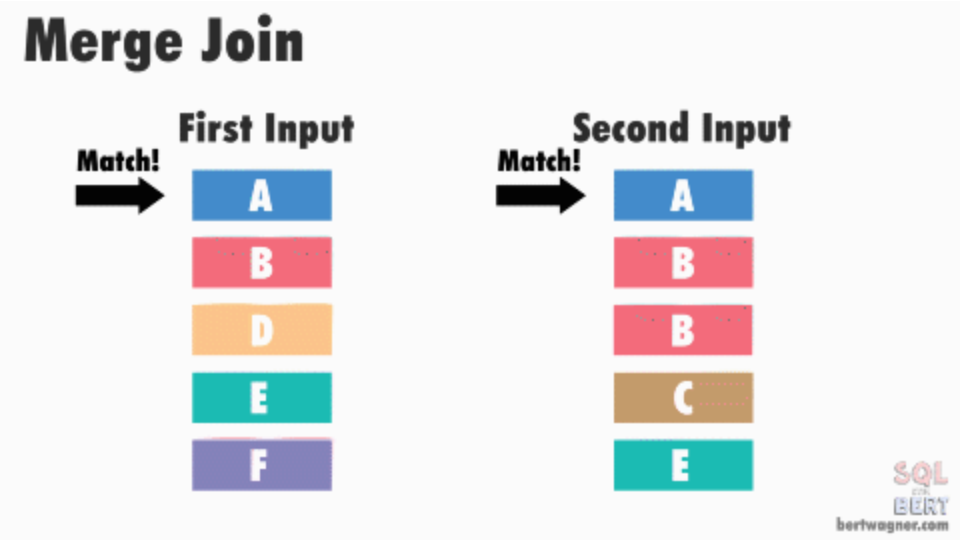


https://bertwagner.com/

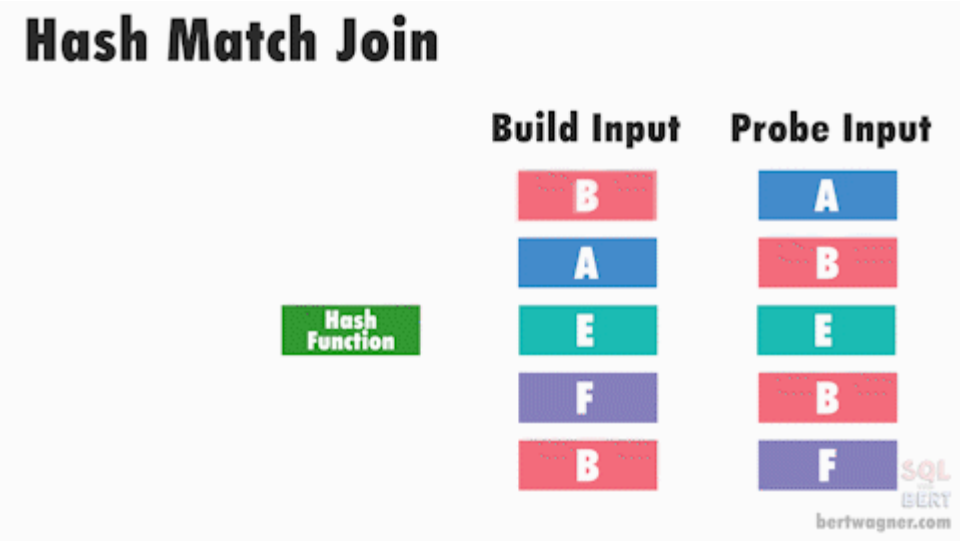
Indexed Nested Loops Join



Merge Join

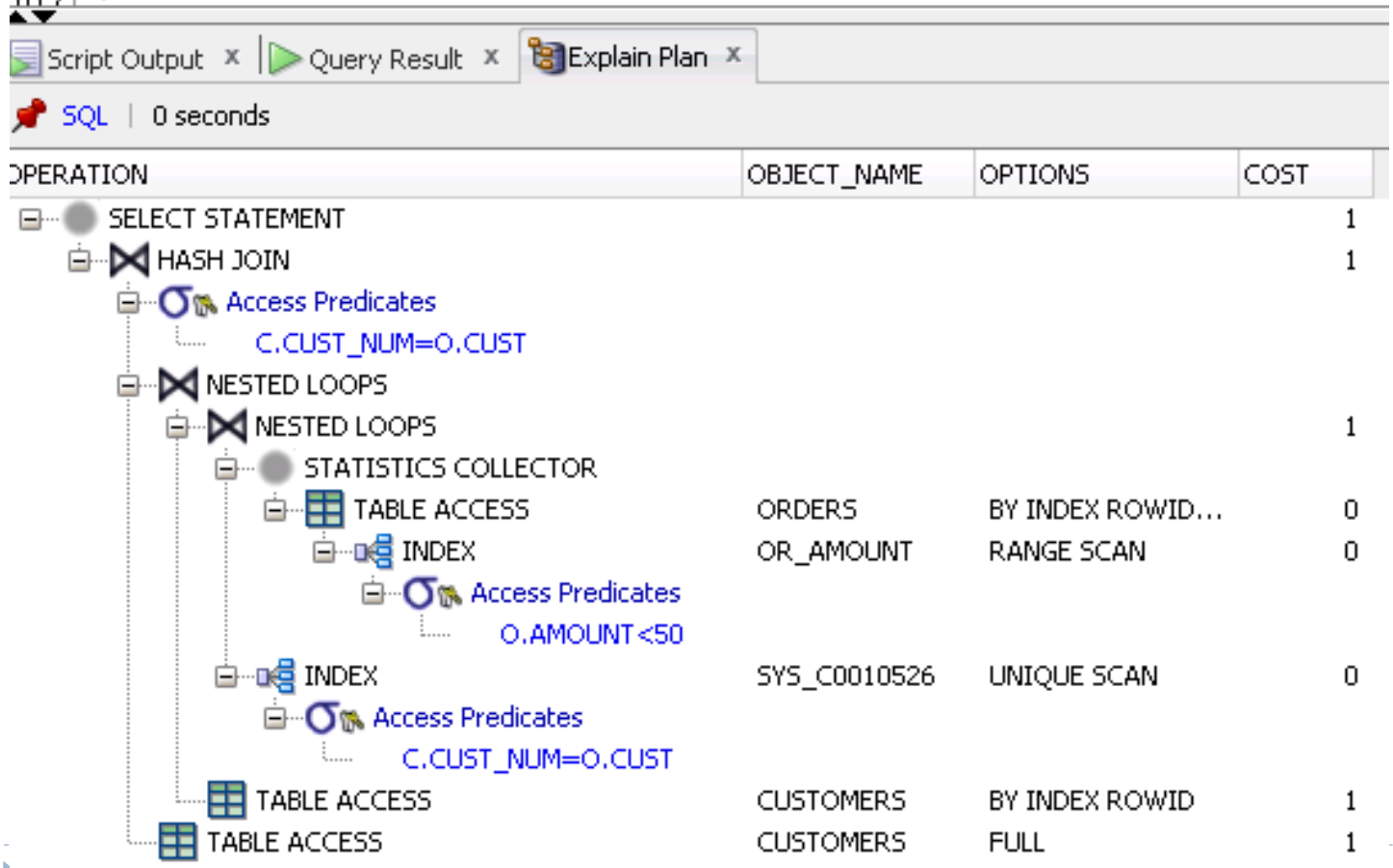

Merge Join
First Input
Match!
A
B
D
E
F
Second Input
Match!
A
B
B
C
E
SQL
BERT
bertwagner.com

Hash Join


Hash Match Join
Build Input
Probe Input
Hash Function
B
A
E
F
B
A
B
E
B
F
SQL
BERT
bertwagner.com

План запроса

```
304  select c.company, o.order_num, o.order_date, o.amount
305  from customers c join orders o on c.cust_num = o.cust
306  where o.amount <50;
307
```



План запроса

```
304  select c.company, o.order_num, o.order_date, o.amount
305  from customers c join orders o on c.cust_num = o.cust
306  where o.amount >50;
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			4
HASH JOIN			4
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	FULL	2
TABLE ACCESS	ORDERS	FULL	2
Filter Predicates			
O.AMOUNT>50			

План запроса

```
311  select c.company, o.order_num, o.order_date, o.amount
312  from customers c join orders o on c.cust_num = o.cust
313  where o.amount < 50 and o.qty >3;
314
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
HASH JOIN			1
Access Predicates			
C.CUST_NUM=O.CUST			
NESTED LOOPS			
NESTED LOOPS			1
STATISTICS COLLECTOR			
TABLE ACCESS	ORDERS	BY INDEX ROWID...	0
Filter Predicates			
O.QTY>3			
INDEX	OR_AMOUNT	RANGE SCAN	0
Access Predicates			
O.AMOUNT<50			
INDEX	SYS_C0010526	UNIQUE SCAN	0
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1
TABLE ACCESS	CUSTOMERS	FULL	1

План запроса

```
311  select c.company, o.order_num, o.order_date, o.amount
312  from customers c join orders o on c.cust_num = o.cust
313  where o.amount < 50 or o.qty >3;
314
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			4
HASH JOIN			4
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	FULL	2
TABLE ACCESS	ORDERS	FULL	2
Filter Predicates			
OR			
O.AMOUNT<50			
O.QTY>3			

План запроса

```
311 create index ind_amount_qty
312 on orders(amount, qty);
313
314 select c.company, o.order_num, o.order_date, o.amount
315 from customers c join orders o on c.cust_num = o.cust
316 where o.amount < 50 and o.qty <3;
317
```

Script Output | Query Result | Explain Plan

SQL | 0 seconds

OPERATION	OBJECT_NAME	OPTIONS	COST
SELECT STATEMENT			1
HASH JOIN			1
Access Predicates			
C.CUST_NUM=O.CUST			
NESTED LOOPS			
NESTED LOOPS			1
STATISTICS COLLECTOR			
TABLE ACCESS	ORDERS	BY INDEX ROWID...	0
INDEX	IND_AMOUNT_QTY	RANGE SCAN	0
Access Predicates			
AND			
O.AMOUNT<50			
O.QTY<3			
Filter Predicates			
O.QTY<3			
INDEX	SYS_C0010526	UNIQUE SCAN	0
Access Predicates			
C.CUST_NUM=O.CUST			
TABLE ACCESS	CUSTOMERS	BY INDEX ROWID	1
TABLE ACCESS	CUSTOMERS	FULL	1

Database Link

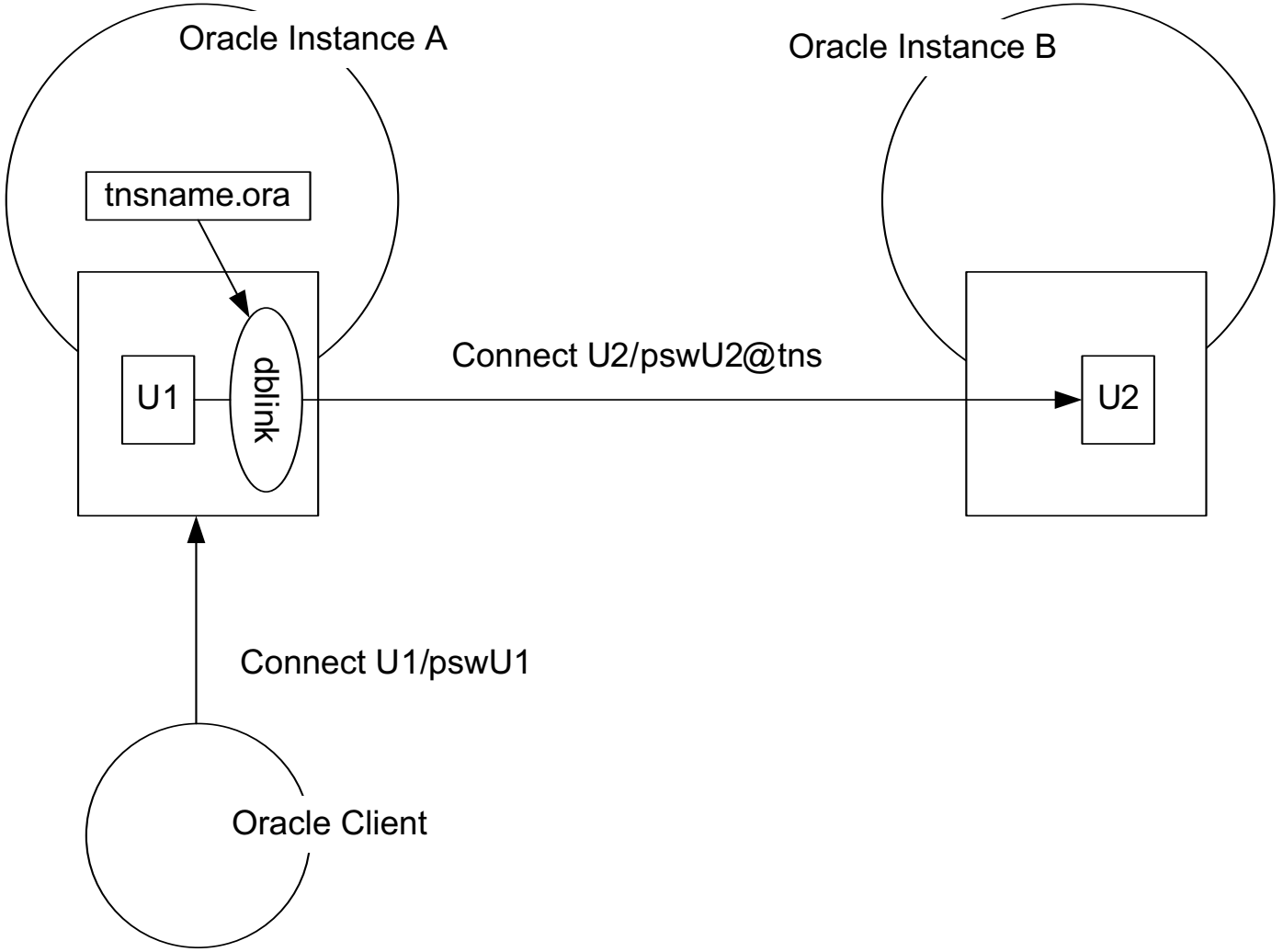
- Database Link (dblink) - объект базы данных, предназначенный для доступа к объектам базы данных, управляемой другим сервером

- Чтобы иметь возможность создать dblink, необходимо выдать привилегии:
- GRANT CREATE DATABASE LINK TO **USERNAME**

Database Link

- Чтобы создать dblink типа user1-user2:
- CREATE DATABASE LINK **anotherdb**
- CONNECT TO **USER2**
- IDENTIFIED BY **PASSWORD**
- USING **'INST_B'**;
- **'INST_B'** - сетевое имя для удаленной БД средствами Oracle Net

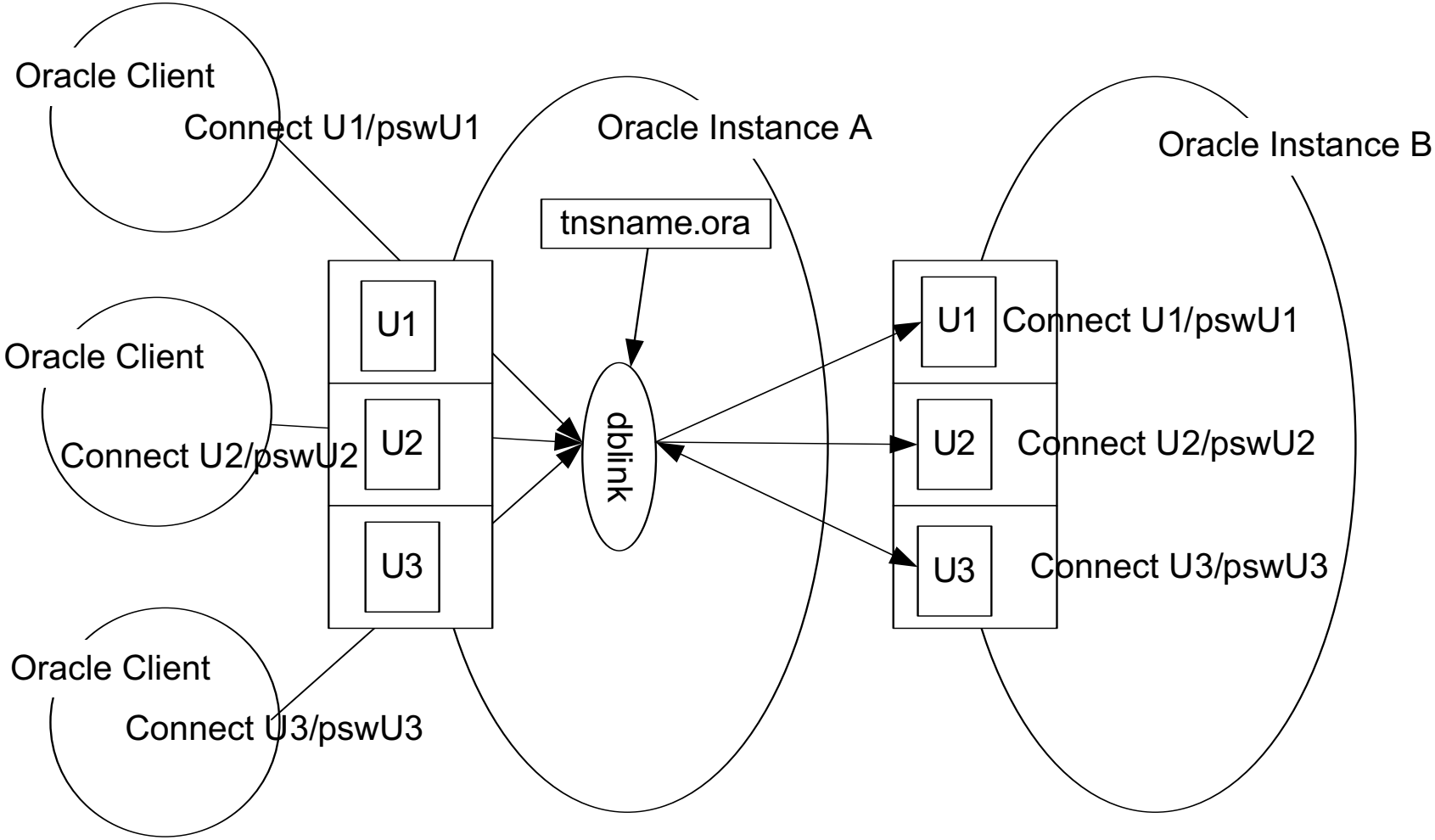
Database Link



PUBLIC DATABASE LINK

- Чтобы иметь возможность создать dblink типа global, необходимо выдать привилегии:
- GRANT CREATE PUBLIC DATABASE LINK,
- DROP PUBLIC DATABASE LINK TO **USERNAME**
-
- Чтобы создать dblink типа global:
- CREATE PUBLIC DATABASE LINK **public_anotherdb**
- USING **'INST_B'**;

PUBLIC DATABASE LINK



DATABASE LINK

- Чтобы обратиться к объектам удаленного сервера, необходимо:
- SELECT name FROM table_name**@anotherdb**;

Database Link

- Чтобы закрыть dblink (открыт на время сессии):
- ALTER SESSION CLOSE DATABASE LINK **anotherdb;**

Database Link

- Словарь для dblink:
- DBA_DB_LINKS
- USER_DB_LINKS
- V$DBLINKS

Database Link

- Параметры Oracle для dblink v$PARAMETER:
- open_links – максимальное кол-во открытых соединений (dblink)в одной сессии.
- open_links_per_instance – максимальное количество соединений для одного экземпляра Oracle.

Вопросы?