

Архитектура **ORACLE**

Объекты базы данных

Лекция 7

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
 - Триггеры
 - Процедуры, функции, пакеты
-



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



Привилегии и роли

- GRANT / REVOKE
- CREATE / ALTER / DROP - DATABASE / USER / PROFILE /
TABLESPACE / ROLE / TABLE / INDEX / TRIGGER /
PROCEDURE / SEQUENCE / VIEW
- WITH ADMIN OPTION
- ANY
- INSERT / UPDATE / DELETE / SELECT / EXECUTE / INDEX /
REFERENCES - имя объекта
- WITH GRANT OPTION
- COLUMN
- CASCADE / RESTRICT



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

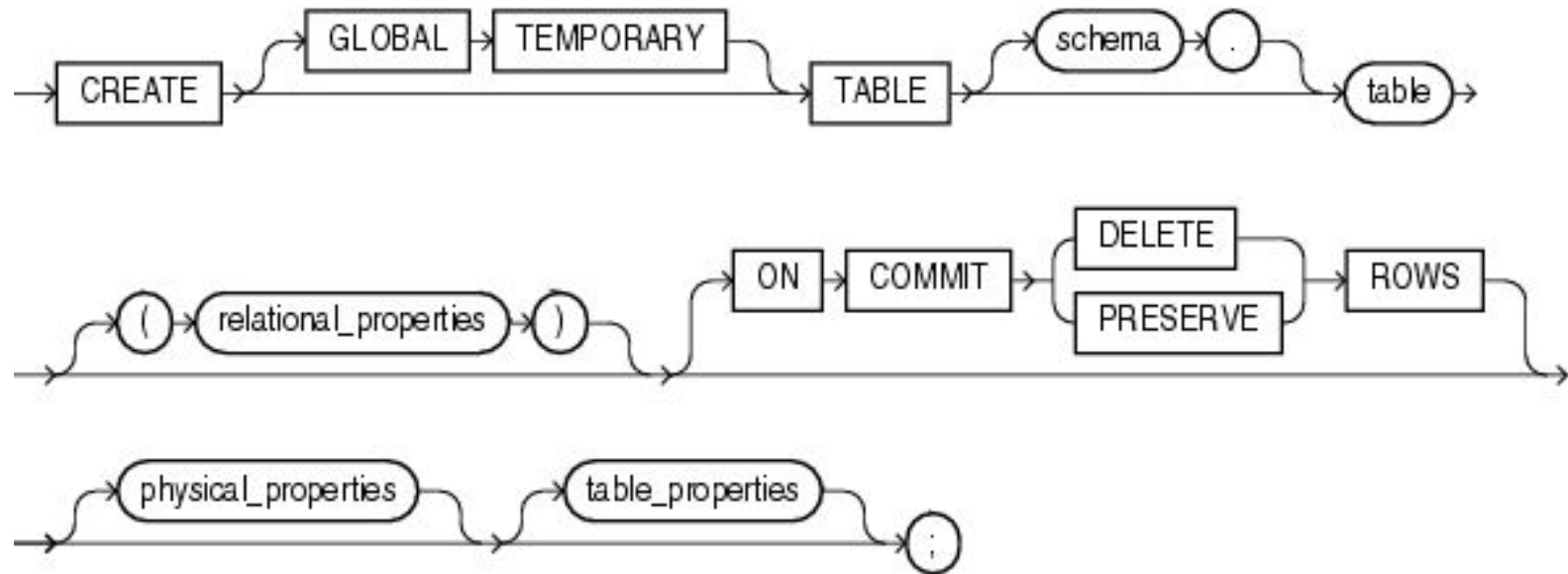


Таблицы

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



Таблицы



Таблицы

```
SVUCORE@sh1> CREATE TABLE SVUCORE.TEACHER
```

```
2  (  
3    TEACHER      CHAR(10) NOT NULL,  
4    TEACHER_NAME VARCHAR2(50),  
5    PULPIT       CHAR(10) NOT NULL  
6  )  
7  /
```

```
CREATE TABLE SVUCORE.TEACHER
```

```
*
```

ошибка в строке 1:

ORA-01542: раздел 'TSDATASVV' явл. автономным, в нем нельзя распределять память

```
SVUCORE@sh1> connect system/system@sh1;
```

Соединено.

sh1 - SYSTEM - 14.10.10

```
SYSTEM@sh1> alter tablespace tsdatasvv online;
```

Раздел изменен.

```
SYSTEM@sh1> connect svucore/svucore@sh1;
```

Соединено.

sh1 - SVUCORE - 14.10.10

```
SVUCORE@sh1> CREATE TABLE SVUCORE.TEACHER
```

```
2  (  
3    TEACHER      CHAR(10) NOT NULL,  
4    TEACHER_NAME VARCHAR2(50),  
5    PULPIT       CHAR(10) NOT NULL  
6  )  
7  /
```

Таблица создана.

```
SVUCORE@sh1> drop table teacher;
```

Таблица удалена.

Параметры

- MONITORING
- LOGGING / NOLOGGING
- ORGANIZATION INDEX

```
SVUCORE@sh1> CREATE TABLE SVUCORE.FACULTY
 2      (
 3          FACULTY          CHAR(10),
 4          FACULTY_NAME     VARCHAR2(50),
 5          CONSTRAINT PK_FACULTY PRIMARY KEY(FACULTY)
 6      )
 7      ORGANIZATION INDEX      -- ???P/INDEX/EXTERNAL
 8      NOMONITORING           -- /MONITORING
 9      TABLESPACE tsdataSVU   -- default:user
10      LOGGING                -- /NOLOGGING
11      PCTTHRESHOLD 20        -- 20% in each block for insert
12  /
```

Параметры PCTFREE и PCTUSED

- Параметр PCTFREE устанавливает процент памяти блока, резервируемой для возможных обновлений строк, уже содержащихся в блоке.
 - Как только блок данных будет заполнен до процента PCTFREE, в этот блок невозможно будет вставить новые строки до тех пор, пока процент памяти, используемой в этом блоке, не упадет ниже значения PCTUSED.
- Параметр PCTUSED задает нижнюю границу, достижение которой вызывает возврат блока данных в список свободных областей.
- Устанавливая разные варианты значений для этих параметров, можно оптимизировать использование дискового пространства.

[illegible]

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

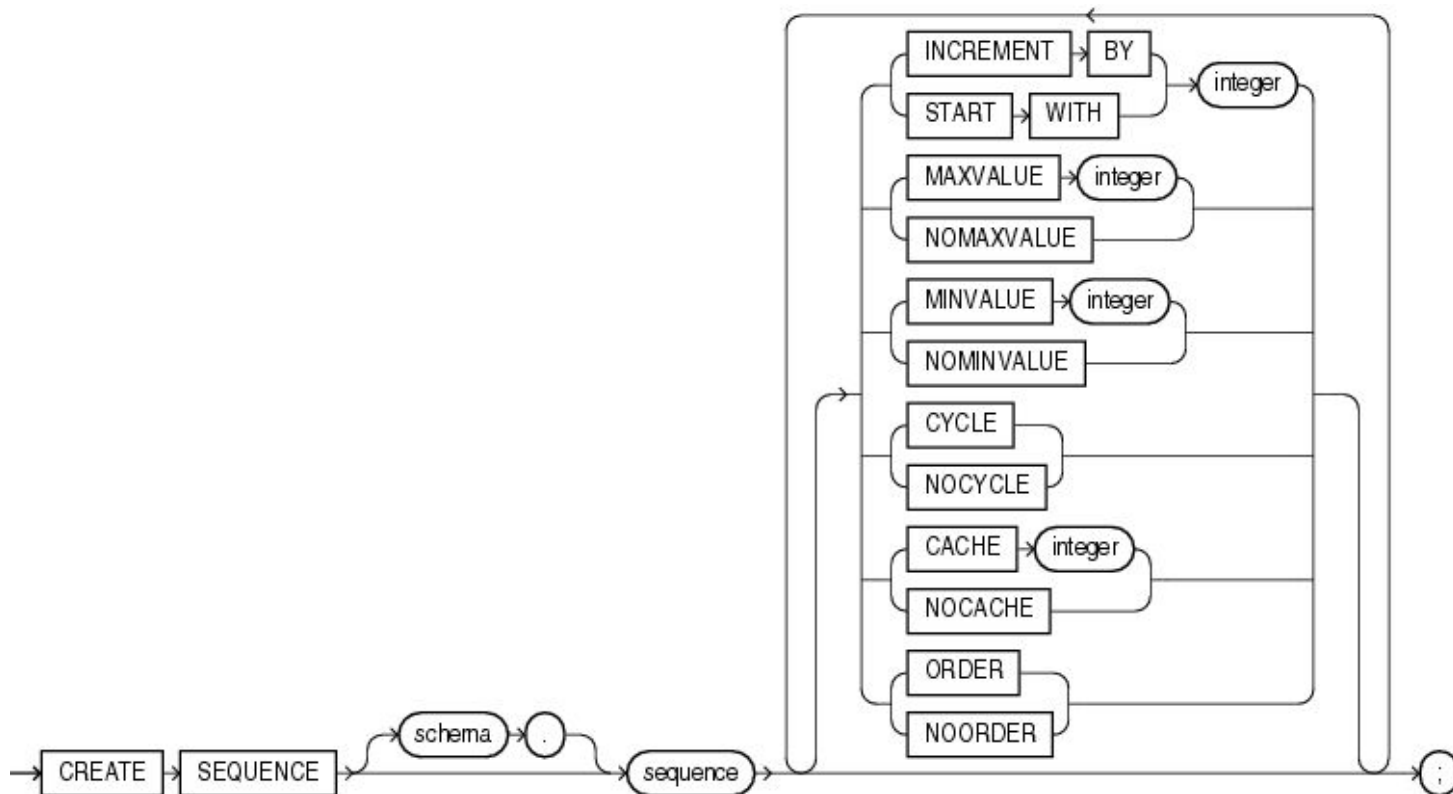


Таблицы

- 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):
 - OOOOOOO – уникальный номер объекта в БД, которому принадлежит строка;
 - FFF – уникальный номер файла данных БД, где хранится строка;
 - BBBBVB – 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

Последовательности

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



Последовательности

□ Привилегия CREATE SEQUENCE

```
-----  
SVUCORE@sh1> create sequence SVUCORE.SCHEDULESSON_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

```
SUVCORE@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
```

```
SUVCORE@sh1> select SHEDULESSON_ID.NEXTVAL from dual;
```

```
      NEXTVAL  
-----  
         100
```

```
SUVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;
```

```
      CURRVAL  
-----  
         100
```

```
SUVCORE@sh1> select SHEDULESSON_ID.NEXTVAL from dual;
```

```
      NEXTVAL  
-----  
         110
```

```
SUVCORE@sh1> select SHEDULESSON_ID.NEXTVAL, SHEDULESSON_ID.NEXTVAL from dual;
```

```
      NEXTVAL      NEXTVAL  
-----  
         120         120
```

```
SUVCORE@sh1> select SHEDULESSON_ID.CURRVAL FROM DUAL;
```

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

Последовательности

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



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 запросы, объединяющие эти таблицы, выполняются быстрее
- Данные сохраняются вместе в кластере, что уменьшает количество операций ввода-вывода и повышает производительность
- Производительность операций вставки, обновления и удаления может быть ниже, чем для обычных таблиц
- Связанные столбцы называются кластерным ключом



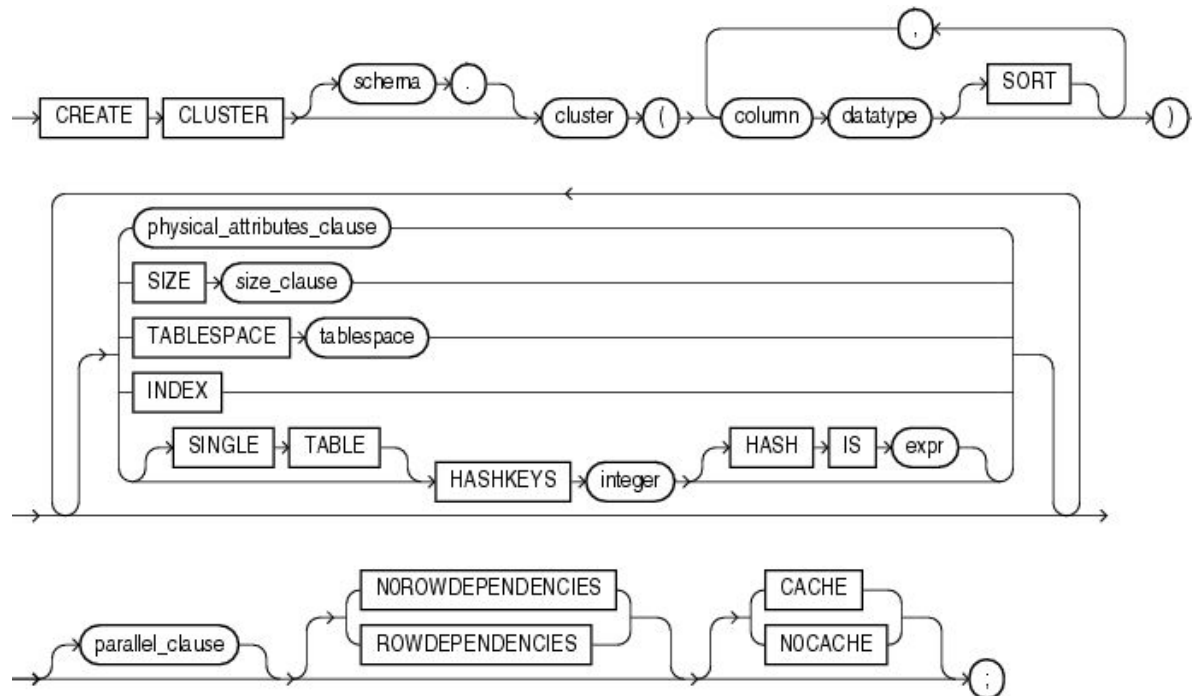
Хэш-кластеры

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

Кластер создан.

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

Таблица создана.

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

Таблица создана.

Кластеры

```
SUUCORE@sh1> 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	SUUCORE	TSDATASUU	HASH	N

11 строк выбрано.

Кластеры

```
SUVCORE@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='TSDATASUU' OR tablespace_name ='TSTEMPSUU'  
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)
TSDATASUU	<u>CLUSTER</u>	1	104	13	851968
TSDATASUU	INDEX	10	80	10	655360
TSDATASUU	TABLE	8	64	8	524288

Хэш-кластеры

```
CREATE CLUSTER HR_cluster (Nom NUMBER(5,0))  
TABLESPACE HR_USERS  
STORAGE  
    (INITIAL 250K  
      NEXT 50K  
      MINEXTENTS 1  
      MAXEXTENTS 3 )  
HASH IS Nom HASHKEYS 200;
```

```
CREATE TABLE HR (Nom NUMBER(5,0) PRIMARY KEY, ...)  
CLUSTER HR_cluster (Nom);
```



Индексы

- Индекс – структура базы данных, используемая сервером для быстрого поиска строки в таблице
- Типы индексов:
 - Табличный (B*Tree) индекс
 - Битовый индекс
 - Функциональный индекс
 - Кластерный индекс



Индексы

- Плотность запроса – количество возвращаемых строк запроса
- Селективность запроса – количество возвращаемых значений ключа (% разных ключей от общего количества)



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



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

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



Индексы

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

Индекс создан.

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

Индекс создан.

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

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

3 строк выбрано.

Синонимы

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



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

- 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

- 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

```
CREATE TABLE SVVCORE.TEACHER
(
  TEACHER      CHAR(10),
  TEACHER_NAME VARCHAR2(50),
  PULPIT       CHAR(10),
  CONSTRAINT PK_TEACHER PRIMARY KEY(TEACHER)
)
```

```
CREATE TABLE SVVCORE.PULPIT
(
  PULPIT      CHAR(10),
  PULPIT_NAME VARCHAR2(100),
  FACULTY     CHAR(10),
  CONSTRAINT PK_PULPIT PRIMARY KEY(PULPIT)
)
CACHE
COMPRESS
```

```
CREATE TABLE SVVCORE.FACULTY
(
  FACULTY      CHAR(10),
  FACULTY_NAME VARCHAR2(50),
  CONSTRAINT PK_FACULTY PRIMARY KEY(FACULTY)
)
ORGANIZATION INDEX
NOMONITORING
LOGGING
```

Материализованные представления

```
DROP MATERIALIZED VIEW COUNT_PFS
```

```
CREATE MATERIALIZED VIEW COUNT_PFS
```

```
BUILD IMMEDIATE
```

```
REFRESH COMPLETE START WITH TO_DATE('07102100','DDMMYYYY') NEXT TO_DATE('14102100','DDMMYYYY')
```

```
AS
```

```
SELECT P.PC, F.FC, T.TC
```

```
FROM (SELECT COUNT(*) PC FROM PULPIT) P,
```

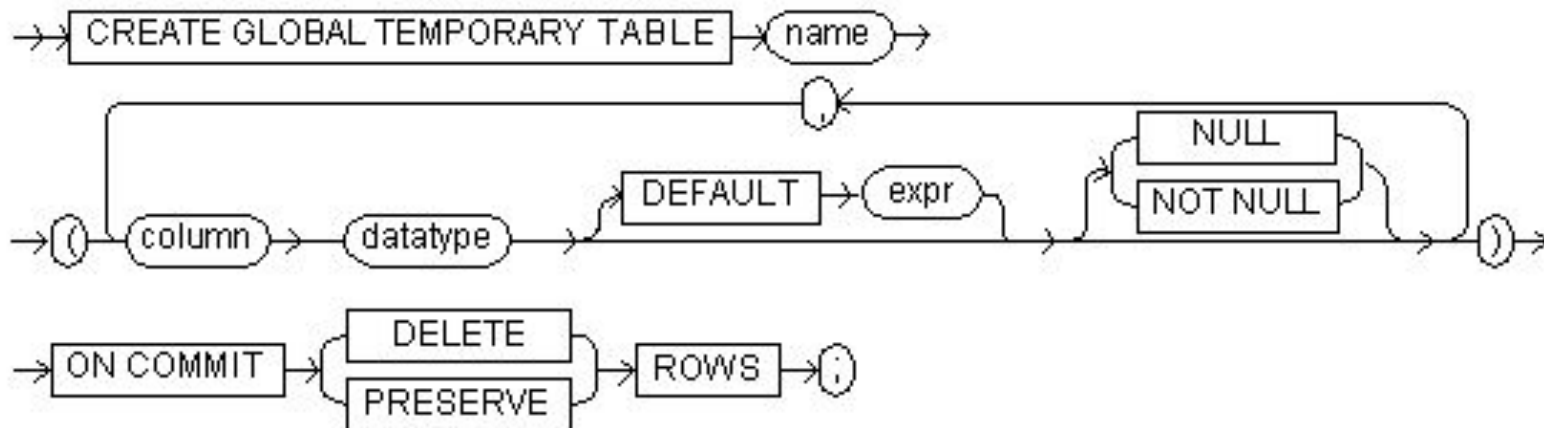
```
(SELECT COUNT(*) FC FROM FACULTY) F,
```

```
(SELECT COUNT(*) TC FROM TEACHER) T
```



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
ORA-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.



Вопросы?

