

8.
Schematy i ich obiekty

Tabele (Tables)

```
create table NAZWA_TABELI
( atrybut typ, [primary key], [unique], [not null], [check]
  atrybut typ [default wartość]
....
....
)
pctfree ...
maxtrans ....
tablespace NAZWA_PRZESTRZENI_TABLICOWEJ
storage
( initial ...
  next ...
  pctincrease ...
  minextents ...
  maxextents ...
);
```

znaczenie wskaźnika wysokiej wody – "high-water mark"

Informacje o tabelach – patrz perspektywy:

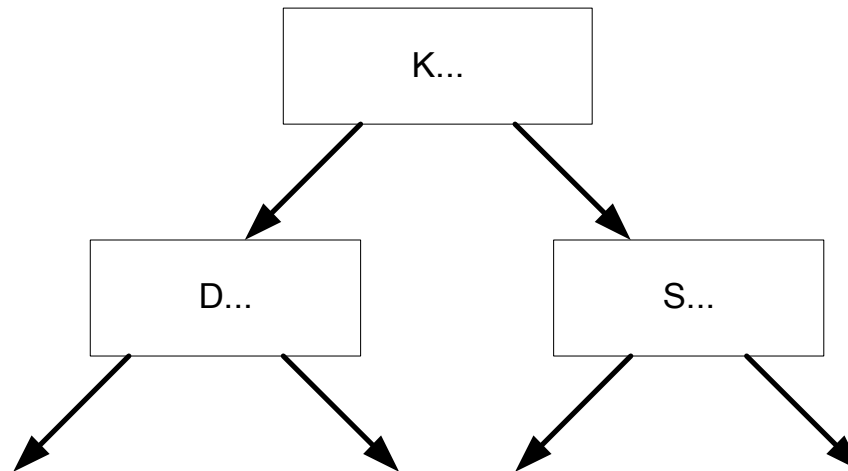
```
DBA_TABLES
ALL_TABLES
USER_TABLES
```

Indeksy (Indexes)

są dodatkowymi zbiorami wskaźników do rekordów uporządkowanymi ze względu na wskazywane wartości
z 1 tabelą można skojarzyć wiele indeksów
indeks można utworzyć zarówno do pustej, jak i do pełnej tabeli

duża liczba indeksów	mała liczba indeksów
szybkie wyszukiwanie	powolne wyszukiwanie
duża zajętość pamięci	oszczędność pamięci
dodatkowe operacje przy zapytaniach typu <i>insert</i> i <i>update</i>	szybkie wstawianie i modyfikacja

Indeksy o strukturze B-drzewa



```

create index NAZWA_INDEKSU
on NAZWA_TABELI
(atrybut, atrybut, atrybut,...)
pctfree ...
...
tablespace NAZWA_PRZESTRZENI_TABLICOWEJ
storage (...);

```

Indeksy bitmapowe

zastosowanie: gdy znamy listę dopuszczalnych wartości w kolumnie, np.

TABELA			INDEKS BITMAPOWY	
Imie	Nazwisko	Płeć	Kobieta	Meczyzna
Jan	Kowalski	Meczyzna	0	1
Piotr	Nowak	Meczyzna	0	1
Joanna	Nowicka	Kobieta	1	0
Adam	Wójcik	Meczyzna	0	1

Zalety indeksów bitmapowych:

mniejsza zajętość pamięci

szybsze tworzenie nowego indeksu

krótszy czas odpowiedzi (szczególnie w przypadku zapytań z licznymi koniunkcjami)

Informacje o indeksach – patrz perspektywy:

DBA_INDEXES
ALL_INDEXES
USER_INDEXES

Klaster (Clusters)

alternatywa do tabel z indeksami, dane przechowywane w sposób uporządkowany

Widoki (Views)

zawężają pole widzenia danych

równoważna nazwa: perspektywy

```
create VIEW studenci_ASI  
as select * from studenci  
where spec='ASI';
```

na widoku można wykonywać dowolne operacje (*select, insert, update*)

Informacje o widokach – patrz tabele:

DBA_OBJECTS
ALL_OBJECTS
USER_OBJECTS

Procedury wbudowane i pakiety funkcji (stored procedures and packages)

Język PL/SQL

instrukcje warunkowe, pętle, zmienne dynamiczne, tworzenie obiektów (kursorów), obsługa wyjątków

Kodowanie i kompilacja:

```
create procedure procedura([lista parametrow])  
as  
[atrybut typ]  
[atrybut typ]  
...  
[kod funkcji]  
/
```

Uruchamianie:

```
SQL> execute nazwa_schematu.nazwa_procedury(parametry);
```

Grupowanie procedur w pakietach:

```
create or replace package NAZWA_PAKIETU  
[lista funkcji (same deklaracje)];
```

```
/  
create or replace package body NAZWA_PAKIETU  
[definicja funkcji 1  
/  
definicja funkcji 2  
/  
definicja funkcji 3  
/];
```

Uruchamianie:

```
SQL> execute nazwa_schematu.nazwa_pakietu.nazwa_procedury_w_pakiecie(parametry);
```

Trigery (Triggers)

funkcje samo-wyzwalające się, „uruchamiane zdarzeniem”, np. na tabeli przy wstawianiu

```
create trigger NAZWA_TRIGERA  
before insert on NAZWA_TABELI  
begin  
...  
end;
```