

# ***Projeto de Banco de Dados***

## ***Turma A-1***

Conteúdo  
Uso de Índices: Oracle

**Professor Rodrigo Salvador Monteiro**  
salvador@ic.uff.br

## ***Exemplo: “Sailors and Boats”***

- Informações devem ser capturadas e armazenadas sobre marinheiros (sailors), barcos (boats) e as reservas que associam um marinheiro a um barco (Reservation)
  - Sailor (id:int, name:string, rating:int, age:int, base:string)
  - Boat (id:int, name:string, colour:string, base:string)
  - Reservation (sid:int, bid:int, day:date)

# Índices Criados Automaticamente

- Em cada relação, o Oracle cria um índice B-tree não clusterizado para a **chave primária**
  - As B-trees se mantêm balanceadas sempre que um elemento é adicionado ou retirado
  - Assim, criaremos índices nas seguintes tabelas e campos:
    - Sailor, id
    - Boat, id
    - Reservations, índice composto em sid, bid e day
- Adicionalmente, o oracle cria um índice para cada **UNIQUE CONSTRAINT**

# Índice Composto

- Se a chave consiste de mais de um campo, o índice é chamado de **índice composto**
  - i.e. um índice para diversos atributos
- Um índice composto é ordenado *lexicograficamente* de acordo com a ordem dos atributos
  - E.g para a chave composta (name,age), teremos:  
 $(\text{Kelly}, 22) < (\text{Kelly}, 63) < (\text{Smith}, 18) < (\text{Smith}, 36)$
  - Lembrar que é diferente do índice (age, name)

# ***Índices Secundários***

- Criar índices secundários para todos os campos que são buscados frequentemente
  - Campos que estão na cláusula WHERE, e não no SELECT
- Neste caso, o usuário tem que criar o índice explicitamente, como será apresentado no próximo slide

# ***Índices Criados Explicitamente***

- Um índice
  - Possui um nome
  - Deve ser criado sobre um conjunto de atributos existentes
  - Pode ser DROPado
- Exemplos
  - `CREATE INDEX sailor_name_idx  
ON Sailor(name);`
  - `DROP INDEX sailor_name_idx;`
  - `CREATE INDEX sailor_name_and_age_idx  
ON Sailor(name, age)`

# ***Utilizando CREATE INDEX (1)***

- O CREATE INDEX cria um índice ordenado sobre a tabela explicitada.
- Uma vez que um índice é criado, ele não é referenciado em uma consulta SQL a não ser que seja para validá-lo (VALIDATE INDEX) ou excluí-lo (DROP INDEX).
- Não podemos criar índice sobre views.
  - Adicionar índices nas tabelas que são base para a view

## ***Utilizando CREATE INDEX (2)***

- O owner de um índice é sempre o mesmo owner da tabela.
- O nome do índice deve ser único para cada owner
  - **Não se pode criar um índice caso a tabela esteja em uso!**
  - O comando CREATE INDEX pode ser extremamente demorado. O SGBD não processa NENHUMA requisição enquanto o índice não tiver sido criado



# ***Overheads vs Desempenho***

- Existe um overhead que deve ser considerado na manutenção de índices secundários.
  - O índice deve ser atualizado sempre que os dados da tabela o forem
  - O índice ocupa espaço em disco
- O DBA deve desempenhar seu papel e tentar achar um “meio termo” entre overhead e desempenho
  - “Faster data retrieval”

# Quando criamos índices secundários (1)

- Adicionar um índice para a chave estrangeira que é frequentemente acessada
  - E.g. bid em Reservations, se nós precisamos frequentemente saber o nome do barco que está em outra tabela
- Adicionar um índice para um atributo qualquer que é usado frequentemente em buscas
  - E.g. day em Reservations (quais reservas temos para hoje ou para amanhã?)
  - E.g. name em Sailor (qual a nota para um marinheiro chamado 'João'?).
  - **Lembrar que pessoas normais pesquisam por nome e não por ID...**

## ***Quando criamos índices secundários (2)***

- Adicionar um índice secundário para atributos referenciados nas cláusulas **order by**, **group by**, **min**, **max**, **avg**
  - E.g. age em Sailor se precisamos saber as idades em ordem crescente
- Adicionar um índice secundário composto que possa prover todos os detalhes frequentemente requisitados pela consulta sem ter que “varrer” a tabela toda
  - E.g. rating e age em Sailor se a query frequente é:  
SELECT rating, AVG(age) FROM Sailor  
GROUP BY rating;

## ***Quando NÃO criar um índice***

- Se a tabela é pequena – poucos registros
- Se a estrutura da tabela é alterada frequentemente
  - drop index,
  - update
  - create index
- Se o atributo que faz parte do índice retorna uma grande proporção de registros
  - E.g. atributo sexo que só pode ser F ou M.

# ***Projeto de Banco de Dados***

## ***Turma A-1***

### Conteúdo Índices Avançados

**Professor Rodrigo Salvador Monteiro**  
salvador@ic.uff.br

# ***Index organised tables***

- Index organised tables
  - Entradas para o índice são registros completos e não somente os row IDs
  - Útil se acessamos a tabela completa frequentemente
  - Para adicionar um índice para uma tabela inteira, adicionar o comando ORGANIZATION INDEX no final do CREATE TABLE

## ***Index organised tables***

- CREATE TABLE admin\_docindex(  
token char(20),  
doc\_id NUMBER,  
token\_frequency NUMBER,  
token\_offsets VARCHAR2(2000),  
CONSTRAINT pk\_admin\_docindex PRIMARY  
KEY (token, doc\_id))  
ORGANIZATION INDEX  
TABLESPACE admin\_tbs  
PCTTHRESHOLD 20  
OVERFLOW TABLESPACE admin\_tbs2;

# ***Índices Bitmap***

- Para cada valor do domínio existe um bitmap que identifica quais row IDs satisfazem as tuplas
- Usualmente aplicados a atributos com domínio esparsos
  - No Oracle, <100 valores distintos
- Atualizar um índice bitmap demora MUITO
- Só use em tabelas que o update, o insert e o delete não seja muito frequente
- Ideal para data warehousing



## ***Exemplo de índice Bitmap***

- Tabela Sailors
- A segunda tabela apresenta um índice bitmap para o campo rating, assumindo valores que variam de 1 a 3

Id	Rating	etc
22	1	Other data
23	2	Other data
31	3	Other data
35	1	Other data

1	2	3
1	0	0
0	1	0
0	0	1
1	0	0

# Clusters

- “Clusters are groups of 1 or more tables physically stored together because they share common columns and are often used together” (Connolly & Begg)
- Melhora o acesso a disco
- As colunas relacionadas das tabelas são chamadas de **cluster key**

## ***Sailor/Reservation cluster***

- Sailor e Reservation podem ser armazenadas em conjunto com o cluster key = Sailor id
- Cada tabela pode ser consultada independentemente
- Quando uma consulta necessitar de uma junção, ela será mais rápida

name	rating	age	id	bid	day
Dustin	7	45	22	101	2/2/08
				103	3/4/08
				101	4/5/08
Rusty	10	35	58	102	4/5/08
				103	6/6/08

## ***Mas devemos criar um cluster para Sailors e Reservations?***

- Um cluster para Sailors e Reservations poderia ser criado caso muitas consultas acessem essas tabelas em conjunto.
- Um cluster para Sailors e Reservations poderia ser ineficiente se na aplicação fossem feitas poucas reservas e muitos cadastros em sailors.

# Clusters Indexados

- Quando usar
  - Consultas que retornam registros sobre um **range** de cluster key values
- E.g.
  - ```
SELECT *  
FROM Sailors S, Reservations R  
WHERE S.id = R.sid  
AND S.id > 50;
```

# ***Definindo um índice de cluster em SQL(1)***

- Primeiramente criar um cluster
  - `CREATE CLUSTER sailor_id_cluster (id int);`
- Definir um índice sobre o cluster
  - `CREATE INDEX sailor_id_idx  
ON CLUSTER sailor_id_cluster;`

## ***Definindo um índice de cluster em SQL(2)***

- Criar tabelas para o cluster
  - CREATE TABLE Sailor (id INT NOT NULL,  
name VARCHAR(20),  
ranking INT,  
age INT)  
CLUSTER sailor\_id\_cluster(id);
  - CREATE TABLE Reservation (sid INT, bid INT.....as  
before....)  
CLUSTER sailor\_id\_cluster(sid);