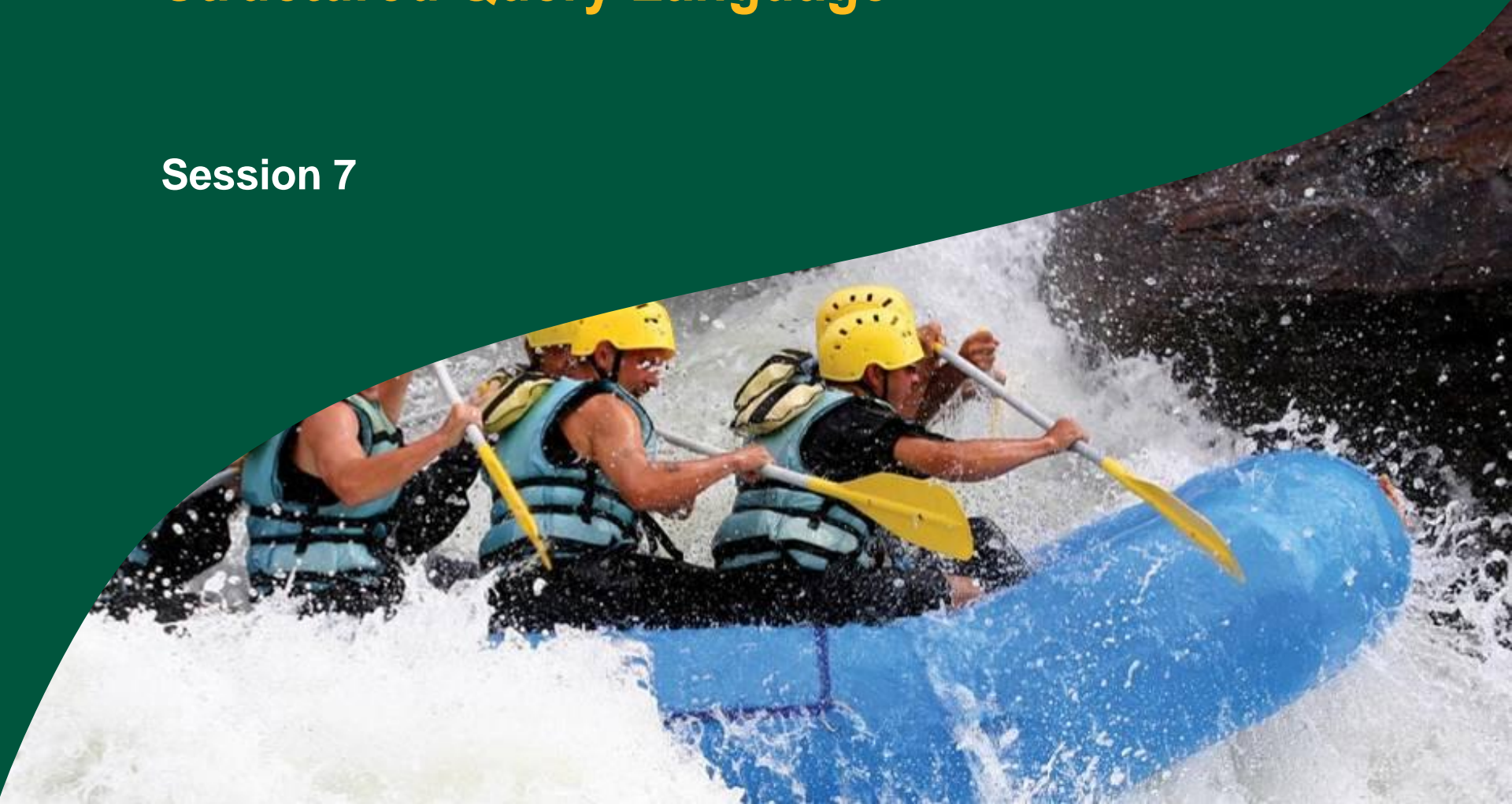


# Structured Query Language

## Session 7



# Coverage

- Working with View
- Working with Indexes
- Working with Sequences
- Using Synonyms
- Controlling User Access

# Working with Views



# Views

- A view is a stored query
- A view takes the output of the query and treats it as a table.
- Used for storing complex queries for easy representation
- Oracle stores the definition of view
- It does not contain data
- It is known as virtual table
- The definition is expanded at runtime when it is used

# Views

**Create or Replace view <viewname> (column\_list)  
as <query>  
with check option constraint  
with READ ONLY**

Create view emp\_view as  
select empno,ename,deptno,sal from emp;

Create view emp\_dept\_view as  
select empno,ename,job,dname  
from emp e, dept d  
where e.deptno = d.deptno

# Views

- Views are derived from base tables and hence have many similarities.
- They can be described and queried
- With some restrictions we can insert into, update or delete data from views
- All the operations are performed on the base tables of the view and they affect the actual data of the base table subject to integrity constraint and triggers

# Views

```
Desc emp_view
```

```
Select * from emp_view
```

```
Insert into emp_view
```

```
Values(101,'Tom',20, 4500);
```



# View Examples

Create view

```
dept_summary(dept_name,emp_count,total_salary,  
maximum_sal, minimum_sal) as  
select dname,count(*),sum(sal),max(sal),min(sal)  
from emp e, dept d  
where d.deptno = e.deptno  
group by dname;
```



# How Views are used

- To provide table level security by restricting data to predetermined rows or columns
- Hides complexity
- Simplifies statements for users
- Save complex queries

# Views – Check option and Read Only

Create view emp\_dept\_10 as

```
select * from emp  
where deptno = 10
```

**with check option**

- DML must confirm to condition specified in where clause

Create OR REPLACE view emp\_dept\_10 as

```
select * from emp  
where deptno = 10
```

**with READ ONLY;**

- DML cannot be performed

# Removing a View

**Drop view <view\_name>**

Drop view dept\_summary;

# Updateable View Restrictions

- A view can be updateable if it does not contain :
  - Set operator
  - Distinct clause
  - Aggregate or Analytical functions
  - Group by clause
  - Subquery in select list
  - Joins (with some exceptions )

# Rules for updateable join view

- The DML statement must affect only one table underlying in the join (known as key-preserved table)
- For an UPDATE statement, all columns updated must be extracted from a key-preserved table.
- For a DELETE statement, the join can have one and only one key-preserved table
- For an INSERT statement, all columns into which values are inserted must come from a key-preserved table

# Updatable view example

Create or replace view emp\_dept\_upd  
as

```
select EMPNO, ENAME, JOB, MGR, HIREDATE, SAL,  
COMM, e.DEPTNO, dname  
From emp e, dept d  
where e.deptno = d.deptno;
```

```
SELECT column_name, updatable  
FROM user_updatable_columns  
WHERE table_name = 'EMP_DEPT_UPD';
```

```
insert into emp_DEPT_UPD  
(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)  
values(1, 'HAPPY', 'CLERK', 7782, '01-JAN-02', 1500, NULL, 10);
```

# Working with Indexes





# Index

- Index helps to locate information faster
- Indexes can be created on column(s) of a table to speed up execution of SQL statements on that table
- Oracle index provides a faster access path to table data
- Indexes are the primary means of reducing disk I/O when properly used.
- A useful tool for application tuning used by developers and DBA's

# Indexes...

- Oracle provides several indexing schemes , the most common used is B-tree structure
- Indexes are automatically created with the same name by Oracle when Primary and Unique constraints are created.
- Oracle maintains and uses indexes on its own
- Columns containing NULL values are not indexed

# Index examples

**Create unique index <index\_name> on  
table(column\_list) asc/desc**

Create index empidx on emp1(empno);

Create index emp\_dept on emp1(deptno,ename);  
create index emp\_job on emp(job)

# When to Create an Index

- A column contains a wide range of values
- A column contains a large number of null values
- One or more columns are frequently used together in a Where clause or a join condition
- The table is large and most queries are expected to retrieve less than 2 to 4 percent of the rows

# When not to create an index

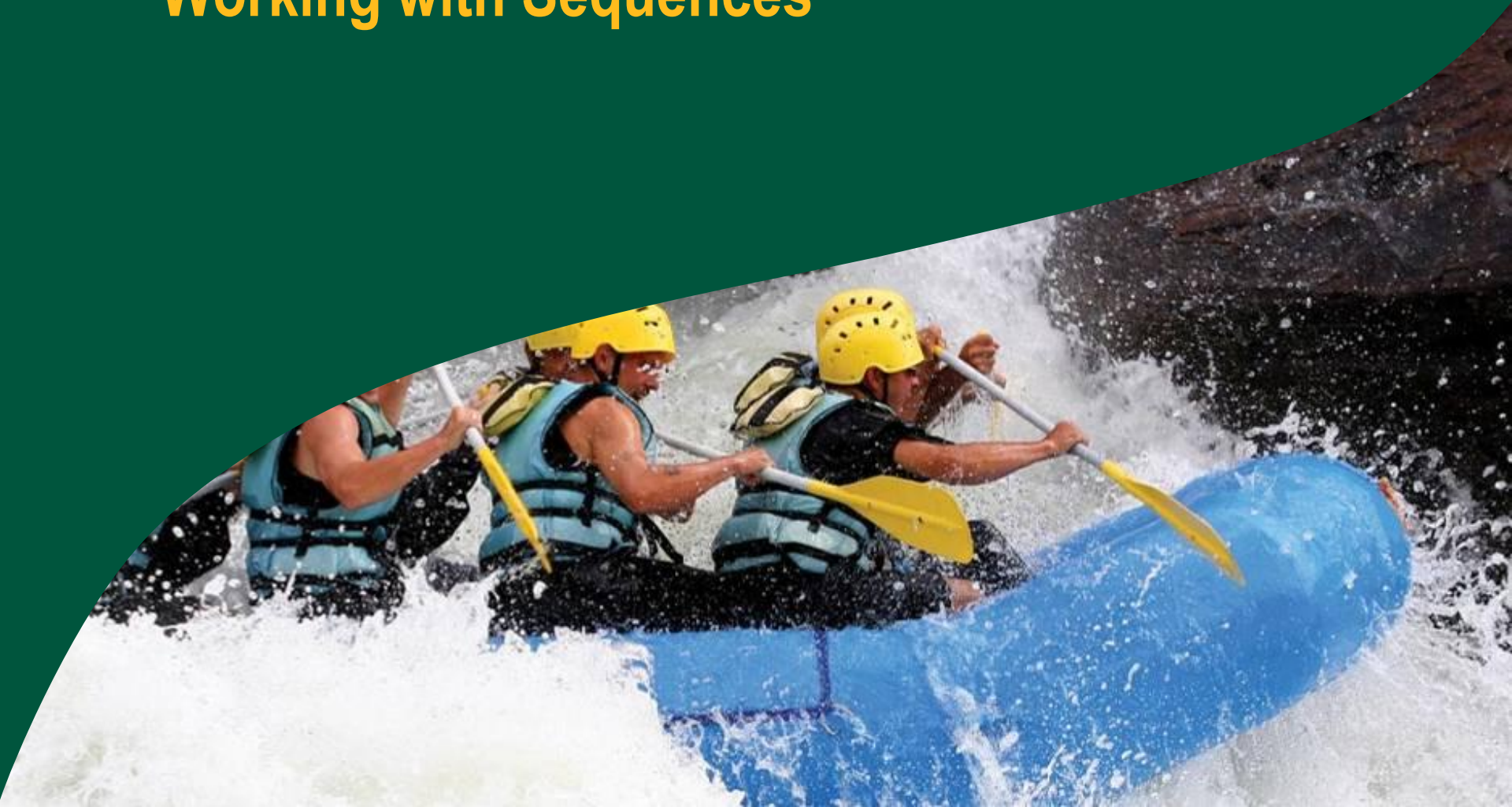
- The table is small
- The columns are not often used as a condition in the query
- Most queries are expected to retrieve more than 2 to 4 percent of the rows in the table
- The table is updated frequently
- The indexed columns are referenced as part of an expression

# Removing an Index

**Drop index <index\_name>**

Drop index empidx;

# Working with Sequences





# Sequences

- Sequences are used for generating unique sequential series of numbers
- Useful in multi-user environment
- Reduces serialization where the statements of two transactions must generate sequential numbers at the same time
- A new sequence number can be generated or the current sequence number can be used by using **NEXTVAL** or **CURVAL**
- They are generated independently of tables
- Used for generating unique primary keys
- The sequence number is incremented independent of transaction committing or rolling back

# Sequence

**Create sequence <sequence\_name>**

**Increment by <value>**

**Start with <value>**

**Maxvalue <value> /nomaxvalue**

**Minvalue <value> / nominvalue**

**Cycle /nocycle**

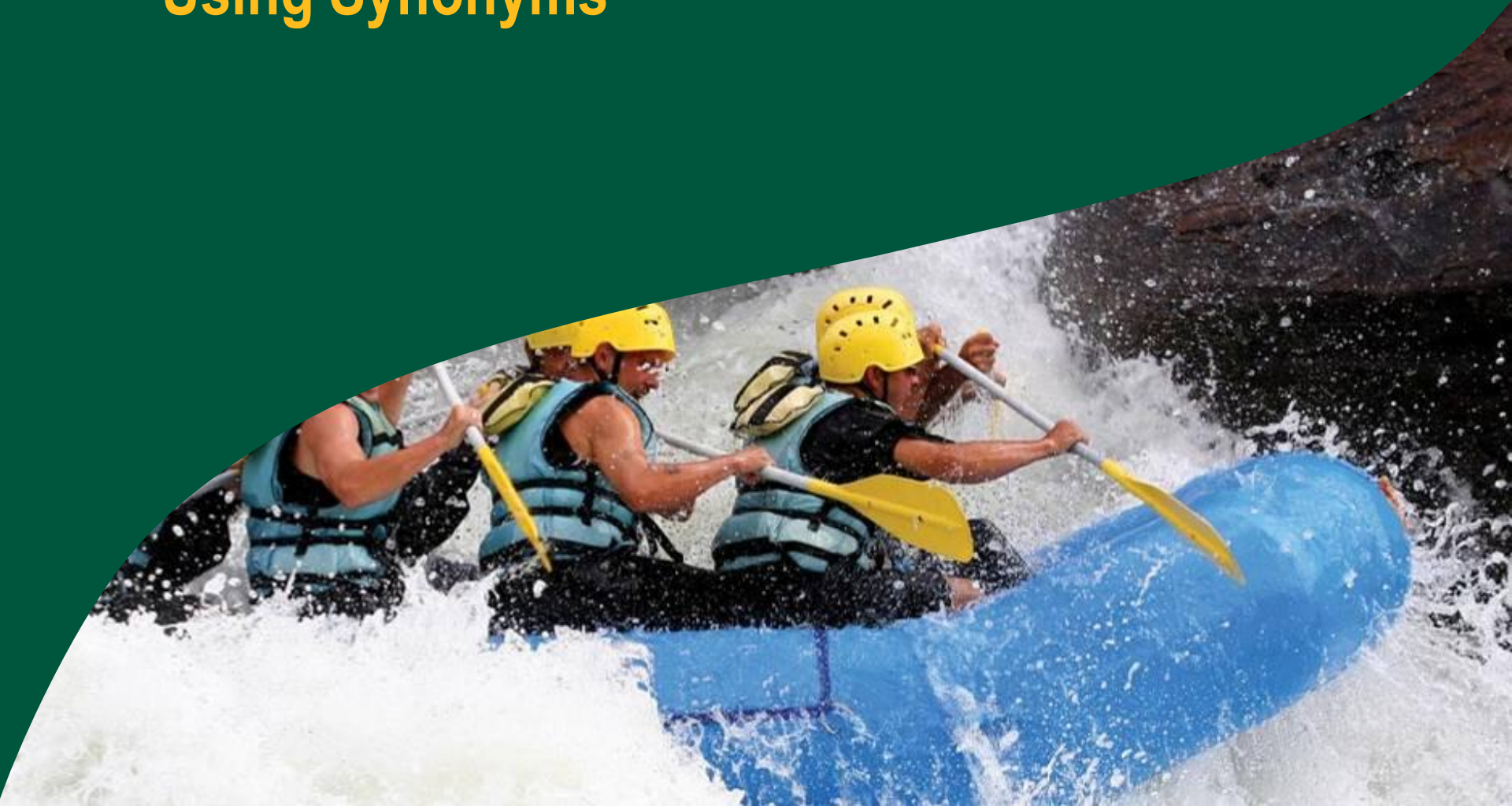
# Sequence example

```
create sequence seq_deptno  
start with 50  
increment by 10  
maxvalue 500 ;
```

```
insert into dept VALUES  
(seq_deptno.NEXTVAL, 'HUMAN RESOURCE', 'NEW  
YORK');
```

```
select seq_deptno.CURRVAL from dual;
```

# Using Synonyms



# Synonyms

- Synonyms are alias name for table,view,sequence,procedures,functions, package,snapshots
- Hides the owner and name of the object
- Provides location transparency in distributed databases.
- Simplifies usage of SQL statements by the users
- Provides data independence
- Synonyms can be Public or Private

# Synonym

**Create [public] synonym <synonym\_name>  
For <object\_name>**

```
create synonym balance for leave_balance;
```

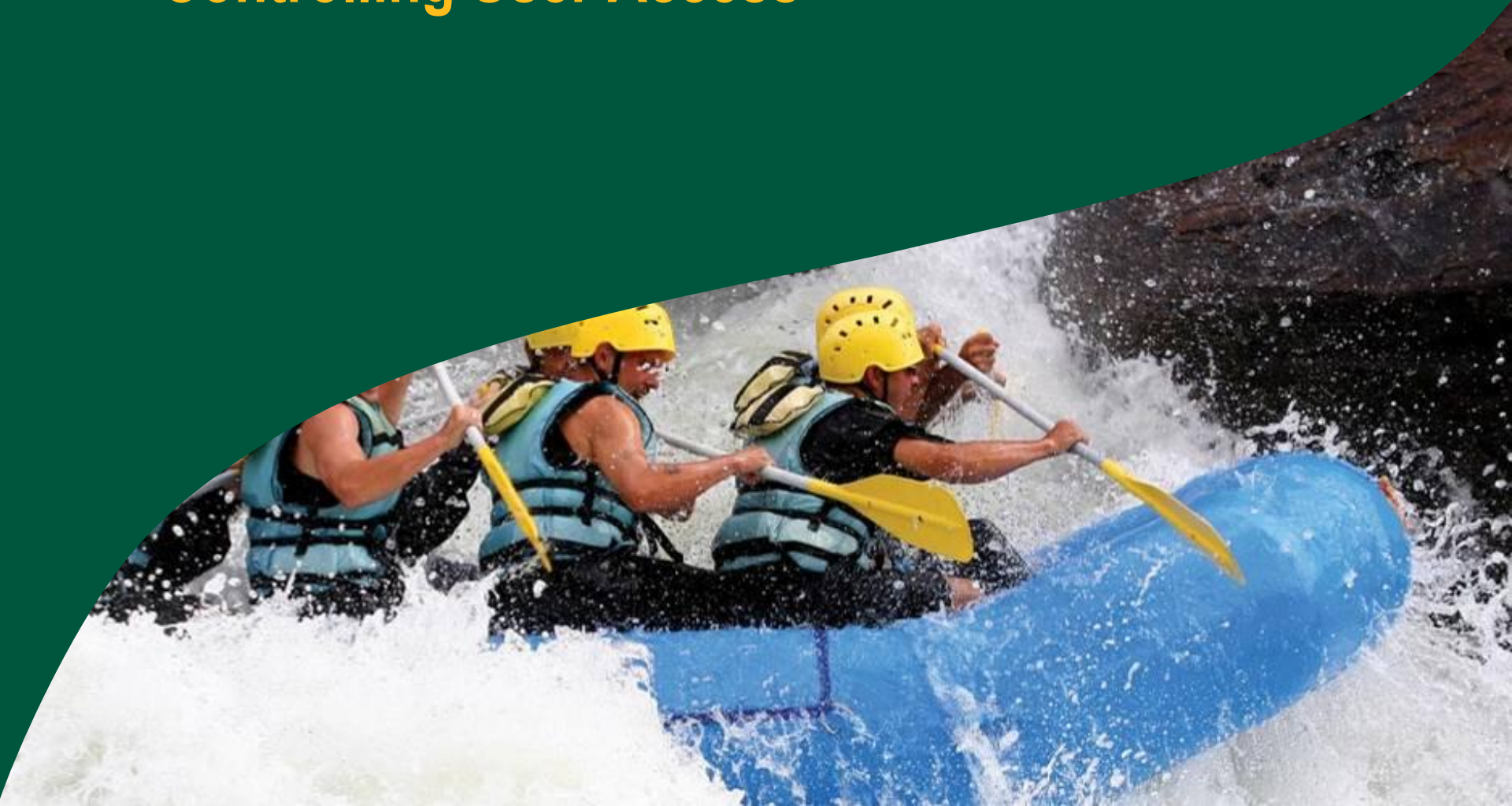
# Removing Synonym

**Drop synonym <synonym\_name>**

Drop synonym balance ;



# Controlling User Access



# Controlling User Access

- Database security can be classified into two categories: system security and data security.
- System security covers access and use of the database at the system level, such as the username and password, the disk space allocated to users, and the system operations that users can perform.
- Database security covers access and use of the database objects and the actions that those users can have on the objects.

# Privileges

- Privileges are the right to execute particular SQL statements
- The database administrator (DBA) is a high-level user with the ability to grant users access to the database and its objects. The users require *system privileges* to gain access to the database and *object privileges* to manipulate the content of the objects in the database.
- Users can also be given the privilege to grant additional privileges to other users

# Creating Users

CREATE USER user IDENTIFIED BY password;

e.g.

CREATE USER scott IDENTIFIED BY tiger;

- The user does not have any privileges at this point.
- You can change your password by using the ALTER USER statement.

e.g.

ALTER USER scott IDENTIFIED BY lion;

# Grant Privileges

- Using Grant command System and Data Privileges can be granted to users or Role.

e.g.

Granting System Privilege

Grant create session to user1;

Granting Data Privilege

Grant select on Emp to user1;

# Role

- A role is a named group of related privileges that can be granted to the user.
- This method makes it easier to revoke and maintain privileges
- A user can have access to several roles, and several users can be assigned the same role.

e.g. CREATE ROLE developer;

- Can Grant privileges to Role .

e.g. GRANT create table, create view TO developer;

- Can Grant Role To Users

e.g. GRANT manager TO user1, user4;

# Using the WITH GRANT OPTION and PUBLIC Keywords

- A privilege that is granted with the WITH GRANT OPTION clause can be passed on to other users and roles by the grantee.

e.g. GRANT select, insert  
ON dept TO user1  
WITH GRANT OPTION;

- An owner of a table can grant access to all users by using the PUBLIC keyword.

e.g. GRANT select  
ON user1.dept  
TO PUBLIC;



# Revoke Object Privileges

- You use the REVOKE statement to revoke privileges granted to other users.
- Privileges granted to others through the WITH GRANT OPTION clause are also revoked.

e.g. REVOKE select, insert  
ON departments  
FROM scott;

# Merge

- Merge statement allows insert or update on a row conditionally. The ON clause is used to specify condition.

MERGE into *table name alias*

Using (*table*) *alias*

On (join condition)

When matched then

update set

col1 =value1,

.

.

coln = valuen

When not matched then

Insert (column1, ....., columnN)

values(val1,.....,valn);

# Merge Example

Merge into emp1 c

Using emp e

On (e.emp\_id=c.emp\_id)

When matched then

update set

c.name=e.name

c.dept\_id=e.dept\_id

.

.

When not matched then

Insert values (e.emp\_id, e.name, e.dept\_id .....);

# Self Study

- Joins
  - Natural Join
  - Inner Join
  - Outer Joins
    - Left
    - Right
    - Full
  - Antijoins
  - Semijoins
- Data Control Language

# Reference

---

- [http://download-uk.oracle.com/docs/cd/B19306\\_01/index.htm](http://download-uk.oracle.com/docs/cd/B19306_01/index.htm)

Thank you !

Kanbay

