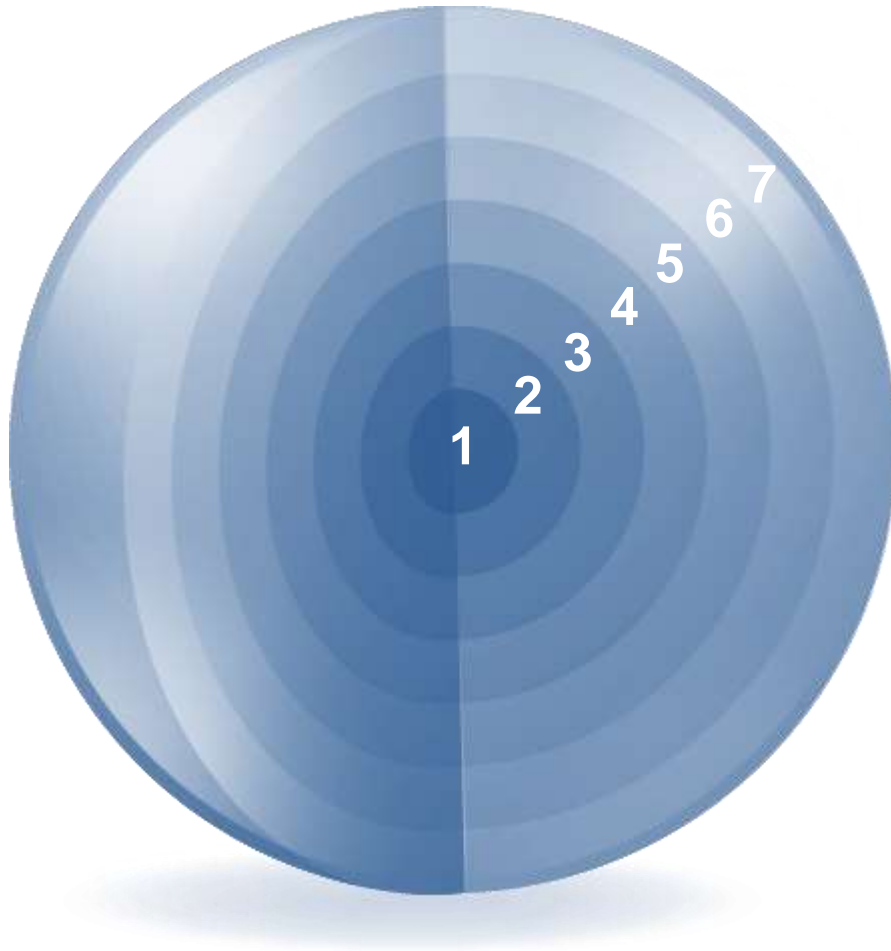


Managing Schema Objects

What You will Learn at the end of this Session?



1. **Alter Table**
2. **Rename Tables**
3. **Comment Table**
4. **Drop columns and set columns as UNUSED**
5. **Drop columns and set columns as UNUSED**
6. **Perform FLASHBACK operations**
7. **Create and use external tables**

- Use the ALTER TABLE statement to:

Add a new column

Modify an existing column

Define a default value for the new column

Drop a column

- Use the ALTER TABLE statement to add, modify, or drop columns:

```
ALTER TABLE table
ADD          (column datatype [DEFAULT expr]
             [, column datatype]...);
```

```
ALTER TABLE table
MODIFY       (column datatype [DEFAULT expr]
             [, column datatype]...);
```

```
ALTER TABLE table
DROP (column [, column] ...);
```

- You use the ADD clause to add columns:

```
ALTER TABLE ord2458  
ADD (Order_completion DATE DEFAULT NULL) ;
```

table ORD2458 altered.

- The new column becomes the last column:

	ORDER_ID	ORDER_DATE	ORDER_STATUS	CUSTOMER_ID	ORDER_COMPLETION
1	2458	20-NOV-99 04.11.54.696211000 AM	0	100	(null)

- You can change a column's data type, size, and default value.

```
ALTER TABLE ord2458  
MODIFY ( Order_id VARCHAR(5) );
```

```
table ORD2458 altered.
```

- A change to the default value affects only subsequent insertions to the table.

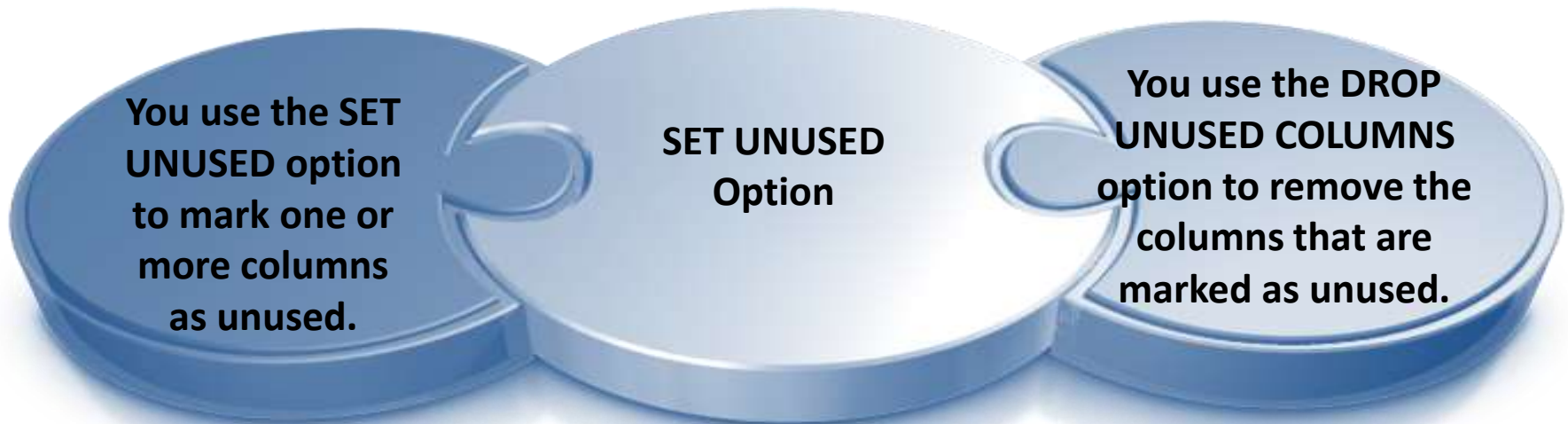
- Use the DROP COLUMN clause to drop columns that you no longer need from the table:

```
ALTER TABLE ord2458  
DROP COLUMN order_completion ;
```

```
table ORD2458 altered.
```

	ORDER_ID	ORDER_DATE	ORDER_STATUS	CUSTOMER_ID
1	2458	20-NOV-99 04.11.54.696211000 AM	0	102

SET UNUSED Option



```
ALTER TABLE <table_name>  
SET UNUSED (<column_name> [ , <column_name>] );  
OR  
ALTER TABLE <table_name>  
SET UNUSED COLUMN <column_name> [ , <column_name>];
```

```
ALTER TABLE <table_name>  
DROP UNUSED COLUMNS;
```



```
DROP TABLE dept80 PURGE ;
```

```
DROP TABLE dept80 succeeded.
```

FLASHBACK TABLE Statement

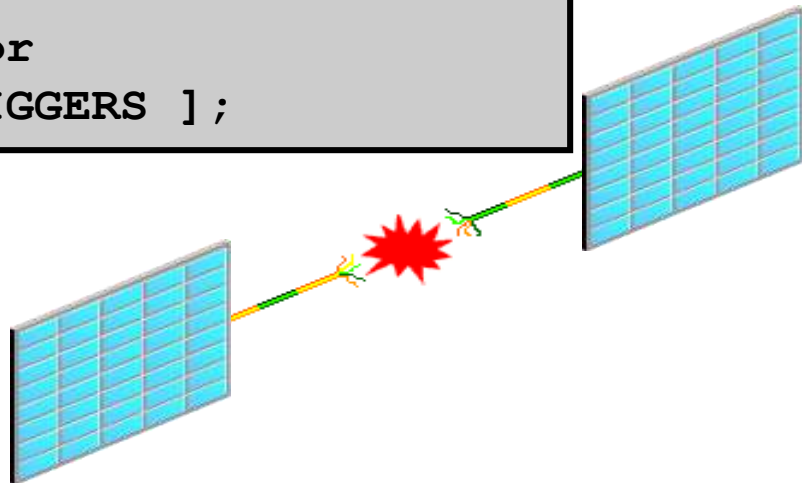
- Enables you to recover tables to a specified point in time with a single statement
- Restores table data along with associated indexes and constraints
- Enables you to revert the table and its contents to a certain point in time or system change number (SCN)



FLASHBACK TABLE Statement

- Repair tool for accidental table modifications
 - Restores a table to an earlier point in time
 - Benefits: Ease of use, availability, and fast execution
 - Is performed in place
- Syntax:

```
FLASHBACK TABLE [schema.] table [,  
[ schema.] table ] ...  
TO { TIMESTAMP | SCN } expr  
[ { ENABLE | DISABLE } TRIGGERS ] ;
```



Using the FLASHBACK TABLE Statement

```
DROP TABLE emp2 ;
```

```
DROP TABLE emp2 succeeded.
```

```
SELECT original_name, operation, droptime FROM recyclebin ;
```

ORIGINAL_NAME	OPERATION	DROPTIME
EMP2	DROP	2009-05-20:18:00:39

...

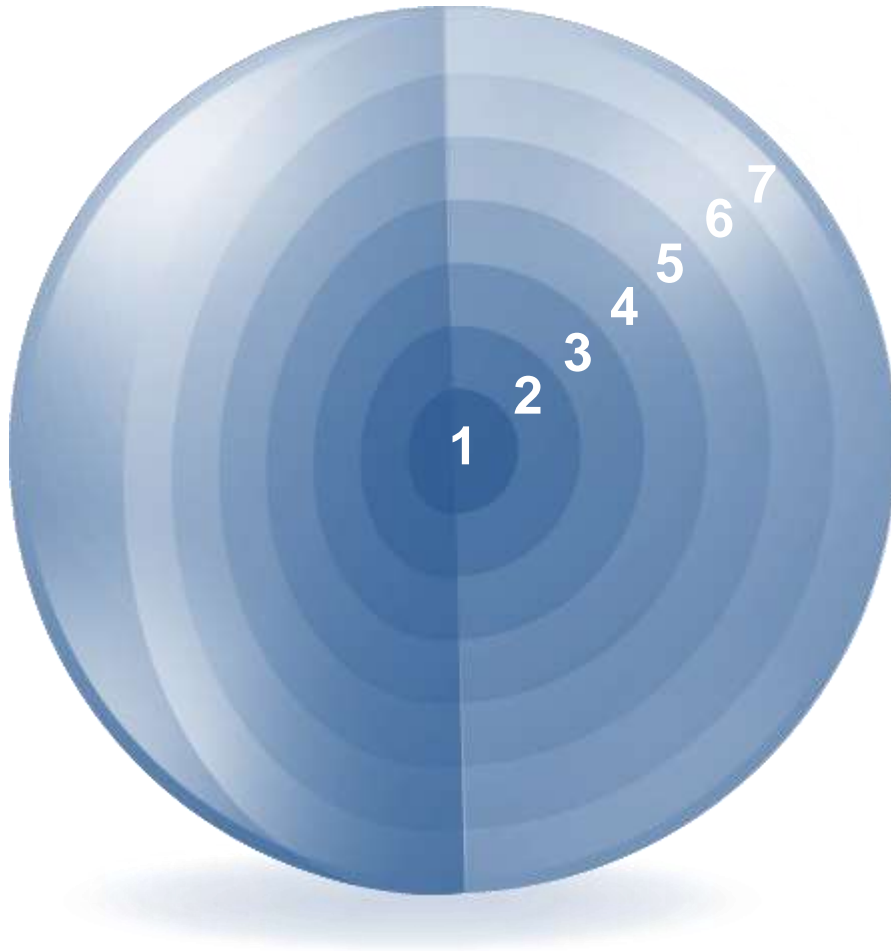
```
FLASHBACK TABLE emp2 TO BEFORE DROP ;
```

```
FLASHBACK TABLE succeeded.
```

•In all the cases, when you execute a DROP TABLE command, the database renames the table and places it in a recycle bin, from where it can later be recovered by using the FLASHBACK TABLE statement.

1.True

2.False



1. Add constraints
2. Create indexes
3. Create indexes by using the **CREATE TABLE** statement
4. Create function-based indexes
5. Drop columns and set columns as **UNUSED**
6. Perform **FLASHBACK** operations
7. Create and use external tables

