Cognixia™

# Oracle SQL

Day 3

# Day 3

- Recap
- Bind Variables
- Nulls
- Pseudocolumns
- Set Operators
  - UNION
  - INTERSECT
  - MINUS

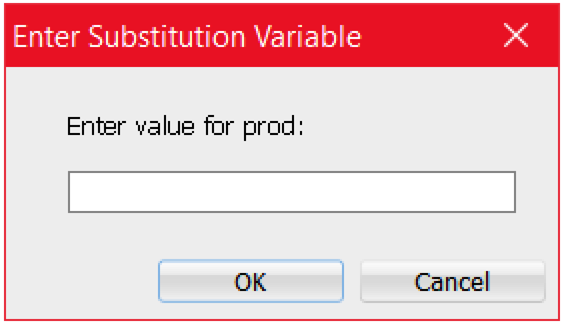
# Bind Variables

Variable whose value is not declared in the statement, but it is requested and substituted at run time. Format : &*variable_name*

E.g.

```
SELECT * FROM OT.PRODUCTS WHERE PRODUCT_NAME =&prod
```

When the query is executed, the user is prompted for a value:



Enter Substitution Variable

Enter value for prod:

The value is then substituted and the query runs.

The bind variable can also be declared first and then used multiple times:

```
DEFINE prod     = &1
SELECT * FROM … WHERE … = &prod
```

Bind variables are said to improve performance in high volume transactional systems. They have negligible effect in data analysis, where the frequency of queries is very low, but could help if a base query is repeated many times by just entering the value instead of modifying the query.

# Null Handling

When executing arithmetic or logical operations on nullable columns, we always have to consider what will happen to a row which has a NULL value.

# Null Functions

Recap the most used functions, with worked examples:

| **Function Name** | **Return Value** |
|---|---|
| COALESCE(*expr1, expr2*, ...) | First value starting from left, which is not NULL. It all expressions resolve to NULL, it return NULL |
| NVL(x,y) | If *'x'* is NULL, replace it with *'y'*. *'x'* and *'y'* must be of the same datatype. |
| DECODE (a, b, c, d, e, ..., default_value) | Checks the value of *'a'*, if *a = b*, then returns *'c'*. If *a = d*, then returns *'e'*. Else, returns *default_value*. |
| NULLIF | |
| | |

# NVL vs COALESCE

NVL is the “old” Oracle implementation, while COALESCE is part of the SQL standard. Most times they work in a similar way, with a couple of important differences:

- COALESCE evaluates multiple options, NVL just one.
- COALESCE could still return NULL, NLV will not.
- COALESCE considers an empty string to be a NULL value

```
select coalesce('','not empty string') from dual;
```

- NVL will do an implicit conversion to the datatype of the first parameter, so the following does not error

```
select nvl('a',sysdate) from dual;
```

- COALESCE expects consistent datatypes, and will throw a 'inconsistent datatype error‘ if not:
  - `select coalesce('a',sysdate) from dual;`

# Null Handling

These statements are equivalent:

```
Select NVL(NULL,'NVL: NULL DETECTED') FROM dual;

Select COALESCE(NULL,'COALESCE: NULL DETECTED') FROM dual;

Select DECODE(NULL,'ha-ha','ho-ho',NULL,'DECODE: NULL
DETECTED','DEFAULT') FROM dual;
```

# Pseudocolumns

Mostly used by operational and technical users, not that useful for data analysis.

ROWNUM, ROWID,  CURRVAL, NEXTVAL, LEVEL

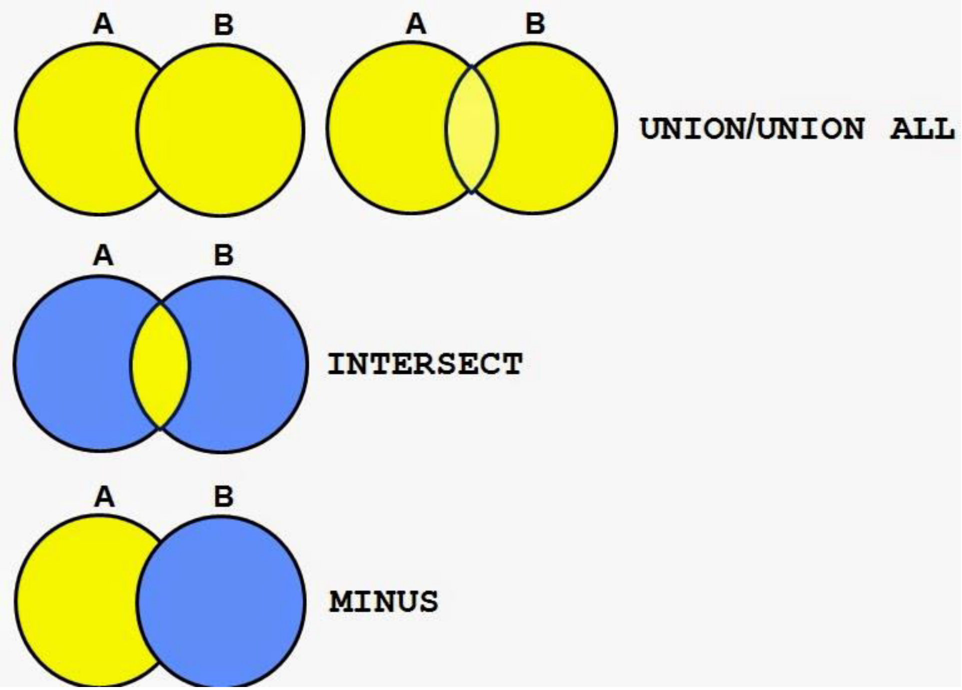
E.g.

- SELECT ROWID, last_name FROM ot.orders;

We will check LEVEL in hierarchical queries

# SET OPERATIONS

Combine multiple queries. All have equal precedence and will be evaluated left-right.



# SET Operators (2)

- Each query  will return its own list of selected columns. These lists must have the same number of elements, be nominated in the same sequence, and be of broadly similar data type. They do not have to have the same names (or column aliases), nor do they need to come from the same tables (or subqueries). If the column names (or aliases) are different, the result set will have columns named as they were in the first query.
- The corresponding columns in the queries must be of the same data type group. The result set of the compound query will have columns with the higher level of precision.
- UNION, MINUS, and INTERSECT will always combine the results sets of the input queries, then sort the results to remove duplicate rows. If the sort order (which is ascending, based on the order in which the columns happen to appear in the select lists) is not the order you want, it is possible to put a single ORDER BY clause at the end of the compound query. UNION ALL is the exception to the sorting-no-duplicates rule: the result sets of the two input queries will be concatenated to form the result of the compound query.

# SET Operator restrictions

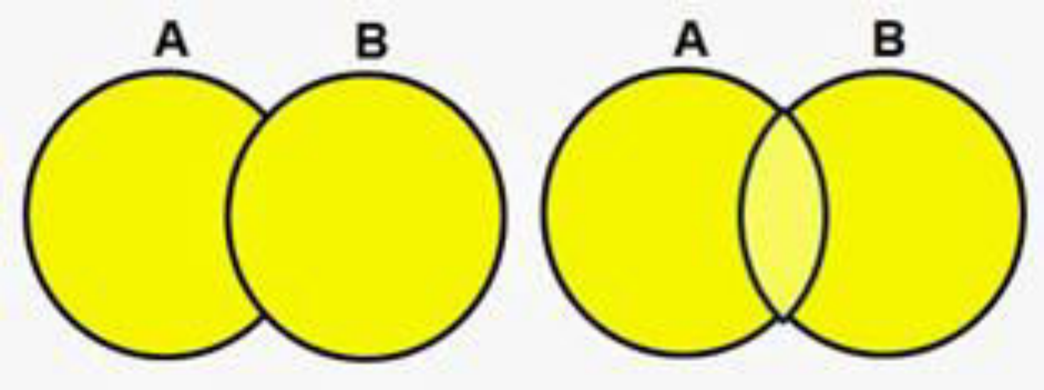
Not valid on columns of type BLOB, CLOB, BFILE, VARRAY, LONG, or nested tables.

If the select list preceding the set operator contains an expression, then one must provide a column **alias** for the expression in order to refer to it in the ORDER BY clause.

One cannot specify the ORDER BY in the subquery of these operators.

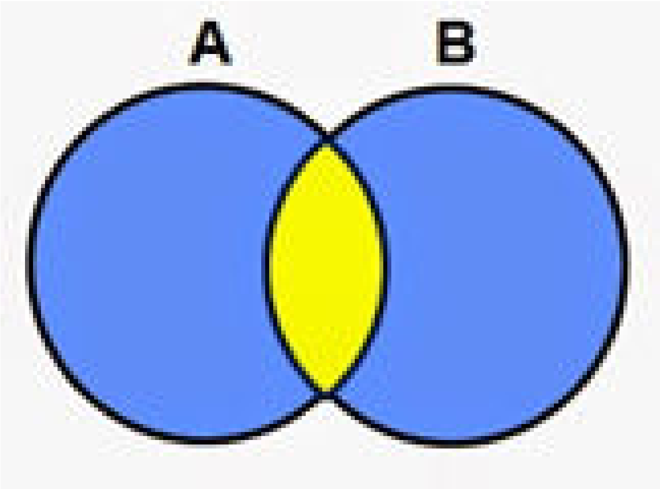
# UNION

Combines records from multiple tables having the same structure.



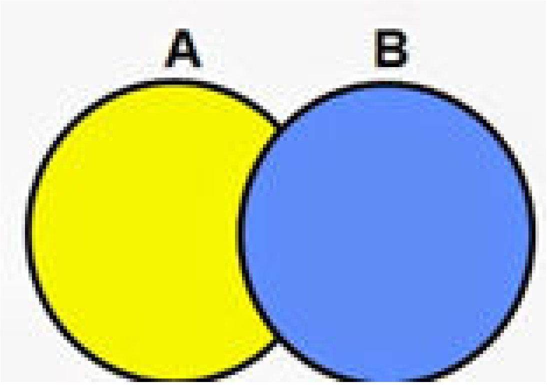
# INTERSECT

Returns the common records from multiple tables having the same structure.



# MINUS

Returns only rows resulted from the first query which do not exist in the second query.



# Constraints

- The entity integrity rule: Primary key attributes don’t permit nulls.
- The referential integrity rule: There mustn’t be any unmatched foreign key values.

# Oracle Dictionary

```
SELECT * FROM DICTIONARY;
```

DBA_TABLES
ALL_TABLES
USER_TABLES

DBA_ view describes all relational tables in the database.
ALL_ view describes all tables accessible to the user.
USER_ view is restricted to tables owned by the user.
Some columns in these views contain statistics that are generated by the DBMS_STATS package or ANALYZE statement.

# Dictionary tables

Oracle captures all metadata and performance statistics in dictionary.

Some interesting tables:

ALL_TABLES, ALL_CONSTRAINTS, ALL_CONS_COLUMNS

# References

| Resource | Location |
|---|---|
| Oracle SQL Reference | |
| Oracle Tables Data Dictionary | https://docs.oracle.com/database/121/ADMIN/tables.htm#ADMIN01508 |
| | |
| | |

Cognixia™

# THANK YOU