[https://tests.mettl.com/authenticateKey/e46500f5](https://tests.mettl.com/authenticateKey/e46500f5" \o "https://tests.mettl.com/authenticatekey/e46500f5" \t "_blank)

<https://tests.mettl.com/authenticateKey/87c41143>

Day3:  
  PBLApp:

Logic building hour plan - M1,M2-A,M2-B  
logic  
dry run  
target code  
mettl -   
demo    
  
WCF assessment:  
MCQ    
Logic Building    
Code Proficiency Checker

john.smith@abc.com  
john.smith\_1@abc.com  
…

<https://tests.mettl.com/authenticateKey/4a72723f>

Collection  
=========  
java.util  
group together obj - single unit  
List

List list = new List(); XXX      
ArrayList      
List list = new ArrayList();        
ArrayList aList1 = new ArrayList();        // non-generic  
ArrayList      <Employee> aList2 = new ArrayList<Employee>();  
indexed        
insertion order        
  
add        
add(obj)          
Employee e = new Employee(100,"John",2000);            
String str ="Java";            
int n = 200;            
double d = 5.6;            
Double dbl = new Double(d);            
aList1.add(e);            
aList1.add(str);            
aList1.add(n);             // int -> Integer  new Integer(n)  
a.List1.add(dbl)            
get        
index based          
int index = 3;            
Object obj = aList1.get(index)            
size        
size()          
Traverse:        
          get() - for   
          enhanced for loop  
for(Object obj :aList1)          //for each Object obj in the list aList1  
{          
//code            
}          
         Iterator:  
Iterator itr = aList1.iterator();          
next()            
remove()            
hasNext()            
ListIterator        
ListIterator listItr = aList1.listIterator();

previous()          
hasPrevious()          
LinkedList      
addFirst()        
addLast()        
  
Vector      
Stack...      
  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
package collndemo;

import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.LinkedList;  
import java.util.List;  
import java.util.Vector;

import emppkg.Employee;

public class TestList {

public static void main(String[] args) {    
List myList = new ArrayList();      
myList.add(123);      
myList.add("Java");      
Employee e=new Employee(1,"John",1000);      
myList.add(e);      
  
List numList = new ArrayList();      
numList.add(12);      
numList.add(123443);      
numList.add(12.4);      
  
List    <String> aList = new ArrayList<String>();  
aList.add("java");      
aList.add("python");      
//aList.add(e);      
  
System.out.println(myList);      
//get      
int index =1;      
int size = myList.size();      
System.out.println("obj at index 1 is "+myList.get(index));      
//traverse      
System.out.println("\*\*\*\*\*normal for loop\*\*\*\*\*\*");      
for(int i=0;i    <size;i++)  
System.out.println(myList.get(i));        
System.out.println("\*\*\*\*\*enhanced for loop\*\*\*\*\*\*");      
for(Object obj : myList)      
System.out.println(obj);        
System.out.println("\*\*\*\*\*iterator\*\*\*\*\*\*");      
Iterator itr = myList.iterator();      
while(itr.hasNext())      
System.out.println(itr.next());        
myList.listIterator().previous();      
  
//new LinkedList    <String>(). //linkList; //new Vector().  
}

}  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
Tomorrow:

Set  
  Map   
  
rdbms/sql/Oracle DBMS

Mercer Mettl | Best Online Assessments Software for Better Talent Measurement, Examinations, Certifications, Recruitment & Training

yes mam

can anyone send friday notes

rdbms

Day5:

rdbms/sql/Oracle DBMS/JDBC to download oracle 115: <https://www.oracle.com/database/technologies/xe-prior-release-downloads.html>

Script for HR schema: <https://raw.githubusercontent.com/connormcd/misc-scripts/master/hr_quick_start.sql>

 Data   
Known Fact File Database       
DataBase Management System  DBMS

Relational Database Management System  RDBMS:    
             Relation - Tabular    
   Table - rows/columns     
 attr/properties       Entity    
Stu    Ex:

   Structured Query Language   SQL:

Not case sensitive    Clauses/Keywords       Commands/Statements

   Data:   
    CHAR - Fixed       char    
     VARCHAR --|\_ varying length     
   VARCHAR2 --|    
 24th Jan 2025           24-jan-25         dd-mon-yy       date    
 Number(5,2)           457.55         Number(p,s)       Number       numeric

    Operation:    
  
Define: Data Definition Language - DDL

   TRUNCATE       DROP       ALTER       CREATE       Define/Redefine/Destroy

     Control: Data Control Language - DCL

 REVOKE       GRANT       privilege- grant/revoke

   MERGE       DELETE       UPDATE       INSERT       manipulate/modify data       Manipulate: Data Manipulation Language - DML

   Fetch/retrieve/search data      - DQL   Retrieve: Data Querying Language

   conn hr/hr   From tablename     col1,col2,...cols | \*   Select    DQL   SELECT … HR: ---

     select \* from tab;

   | run - services.msc   |\_\_\_ server instance - XE | ORCL gui - sql developer    ---------- cli - sqlplus      server -----------  -    client  …. Oracle ======    
ListnerService run - cmd - lsnrctl ----------    OracleService

SELECT col | Exp | function FROM tablename(s) [WHERE]   
-- filter [GROUP BY]       
-- grouping attr [HAVING]       
     aggregation     multi   single   -- filter based agg result [ORDER BY] - sort - arrange asc/desc Functions       
count,sum,min,max,...

  Describe | Desc      Structure:

column nullable? datatype    Desc Employees

 Q) Display the employee details:   
Output: Employee ID, Name, Salary

From Employees;  Select Employee\_id,First\_name,Salary

as- alt title/alias name

From Employees;  Select Employee\_id "Employee ID",First\_name as Name ,Salary

or

Select Employee\_id as "Employee ID",First\_name "Name" ,Salary from Employees;

Q) Display the annual salary of employees of department 50

o/p: empid,name,sal,annSal

'10-jan-25'  'java'   Q)display the name by appending first name and last name with a space char/date lit : 'val'

john smith    first\_name || ' ' || last\_name   operator to append: ||

salary \* 12 annsal From Employees Where department\_id = 50  salary sal,   first\_name || ' ' || last\_name as Name,   employee\_id as empid,   select

cond1 and ( cond2 or cond3 )    or cond2 more than 2 cond   cond1    and cond2   cond1     Where: more than 1 cond

Q) Display the annual salary of employees of department 50.   
Arrange the records based on their annual salary o/p: empid,sal,annSal,dno

department\_id dno From Employees Where department\_id = 50 Order by annsal  salary \* 12 annsal,   salary sal,   first\_name || ' ' || last\_name as Name,   employee\_id as empid,   select

 department\_id dno From Employees Where department\_id = 50 Order by salary\*12   salary \* 12 annsal,   salary sal,   first\_name || ' ' || last\_name as Name,   employee\_id as empid,   or select

department\_id dno From Employees Where department\_id = 50 Order by 4  salary \* 12 annsal,   salary sal,   first\_name || ' ' || last\_name as Name,   employee\_id as empid,   or select

asc default/desc    pos | exp | alias   Order By: col

Q)Display Employees who are earning a salary above 5000. Arrange the records based on their department id and job id  
o/p: department\_id,job\_id,eid,salary

salary From Employees Where salary  employee\_id as eid,   job\_id,   department\_id,   Select > 5000 Order by department\_id, job\_id

Distinct - distinct records    ---Oracle DISTINCT clause is used to remove the duplicate records from the result set--

from employees;  select distinct department\_id,job\_id

Q) Display the department wise average salary   
sum,count,min,max,avg  o/p: dno,avgsal agg functions:

 select avg(salary) avgsal from employees;

107 emp - avgsal avgsal ------- 324354

   select department\_id, avg(salary) avgsal from employees group by department\_id dno

124235    1232 20    ------- 10    avgsal ----

select department\_id,round(avg(salary)) from employees where department\_id = 10 or department\_id = 30 group by depar tment\_id order by 1;

Q) Display the department wise and job wise average/total/minimum/maximum salary

o/p: dno,avgsal,totalsal,minsal,maxsal

select department\_id, job\_id, avg(salary) avgsal,sum(salary) totalsal, min(salary) minsal,max(salary) maxsal from employees group by department\_id,job\_id order by 1,2

/

count() row count count() all rows count(col) all non-null values of the col --

SQL> select count() from employees; COUNT(\*)

---------- 107

SQL> select count(employee\_id) from employees;

COUNT(EMPLOYEE\_ID) ------------------ 107

SQL> select count(commission\_pct) from employees;

COUNT(COMMISSION\_PCT) --------------------- 35

SQL> select count(employee\_id) from employees where commission\_pct is not null; COUNT(EMPLOYEE\_ID) ------------------ 35

SQL> select count(employee\_id) from employees where commission\_pct is null; COUNT(EMPLOYEE\_ID) ------------------ 72

SQL> select count(job\_id) from employees; COUNT(JOB\_ID) ------------- 107

SQL> select count(distinct job\_id) from employees;

COUNT(DISTINCTJOB\_ID) --------------------- 19 -----------

especially for agg result based cond  filter   Having:

Q) Display department id and employee count if the count is greater than 10

 o/p: dno,empcount

Having count(employee\_id)  Group by department\_id   From employees   Select department\_id dno, count(employee\_id) empcount   > Order by 1  10

Q) Display department id and employee count if the count is less than 10   
o/p: dno,empcount

department\_id dno, count(employee\_id) empcount From employees Where department\_id is not null Group by department\_id Having count(employee\_id)  Select < 10 Order by 1

 Employees Department n table to join , there must n-1 join condition table1 table2 two types   / Join :

  inner - matched

dno,dname  eid,name,sal,dno departments:   outer - matched/unmatched employees:

based on all common attr between tables  natural join:

select \* from Employees natural join departments

Inner join:  
Inner Join is the simplest and most common type of join. It is also known as simple join. It returns all rows from multiple tables where the join condition is met.

 on employees.dno = departments.dno   select eid,name,sal,employees.dno,dname from employees inner join departments

or

using(dno)  select eid,name,sal,employees.dno,dname from employees inner join departments

left outer | right outer | full outer left outer:  Outer join:

  on employees.dno = departments.dno   select eid,name,sal,employees.dno,dname from employees left outer join departments

or

using(dno)  select eid,name,sal,employees.dno,dname from employees leftouter join departments

right outer:

on employees.dno = departments.dno  select eid,name,sal,employees.dno,dname from employees right outer join departments

or

using(dno)  select eid,name,sal,employees.dno,dname from employees right outer join departments

full outer:

 on employees.dno = departments.dno   select eid,name,sal,employees.dno,dname from employees full outer join departments

or

using(dno)  select eid,name,sal,employees.dno,dname from employees full outer join departments

Q) Display employee details with their department details o/p: eid,name,dname,sal eid,name,sal -> employees dname -> departments

Q) Display location-wise average salary of employees o/p: city dname avgsal city -> locations dname -> departments avgsal - employees

Retrieve    Manipulate     Control     Monday: DDL DML JDBCefine