COURSE PLAN

Department : DATA SCIENCE & COMPUTER APPLICATIONS

Course Name & code : Database Lab & DSE 2260

Semester & branch : **4th** Semester, BTech

Name of the faculty : Vinayak Mantoor & Archana. H

No. of contact hours/week : 3 hours/week

Continuous Evaluation	60%
Four Evaluations	10 marks-obs (3) + Exe (2) + Test (5) 10 marks- obs (3) + Exe (2) + Test (5) 10 marks- obs (3) + Exe (2) + Test (5) 10 marks- obs (3) + Exe (2) + Test (5) 10 marks- obs (3) + Exe (2) + Test (5) 10 marks mini project **Max.Marks:60**
Lab Examination	**40%**
	Examination of 3 hours duration that includes questions based on: SQL (20 marks approx.) + PL/SQL (20 marks approx.) Max.Marks:40

INSTRUCTIONS TO STUDENTS

1. Students should be regular and come prepared for the lab practice.
2. In case a student misses a class, it is his/her responsibility to complete that missed exercise(s).
3. Students should bring the observation book daily for the lab.
4. They should implement the given query/program individually.
5. Students should clearly listen to the instructions given by the faculty.
6. Once the query/program gets executed, they should show the query/program and results to the instructors and copy the same in their observation book.

7. When copying down the query in the observation book the template to be followed is:
    a. Question No
    b. Question
    c. Query statement
    d. Output
8. When copying down the PL/SQL program in the observation book the template to be followed is:
    a. Program No
    b. Program title
    c. Program Code
    d. Output
9. Questions for lab tests and exam need not necessarily be limited to the questions in the manual, but could involve some variations and / or combinations of the questions.

CONTENTS

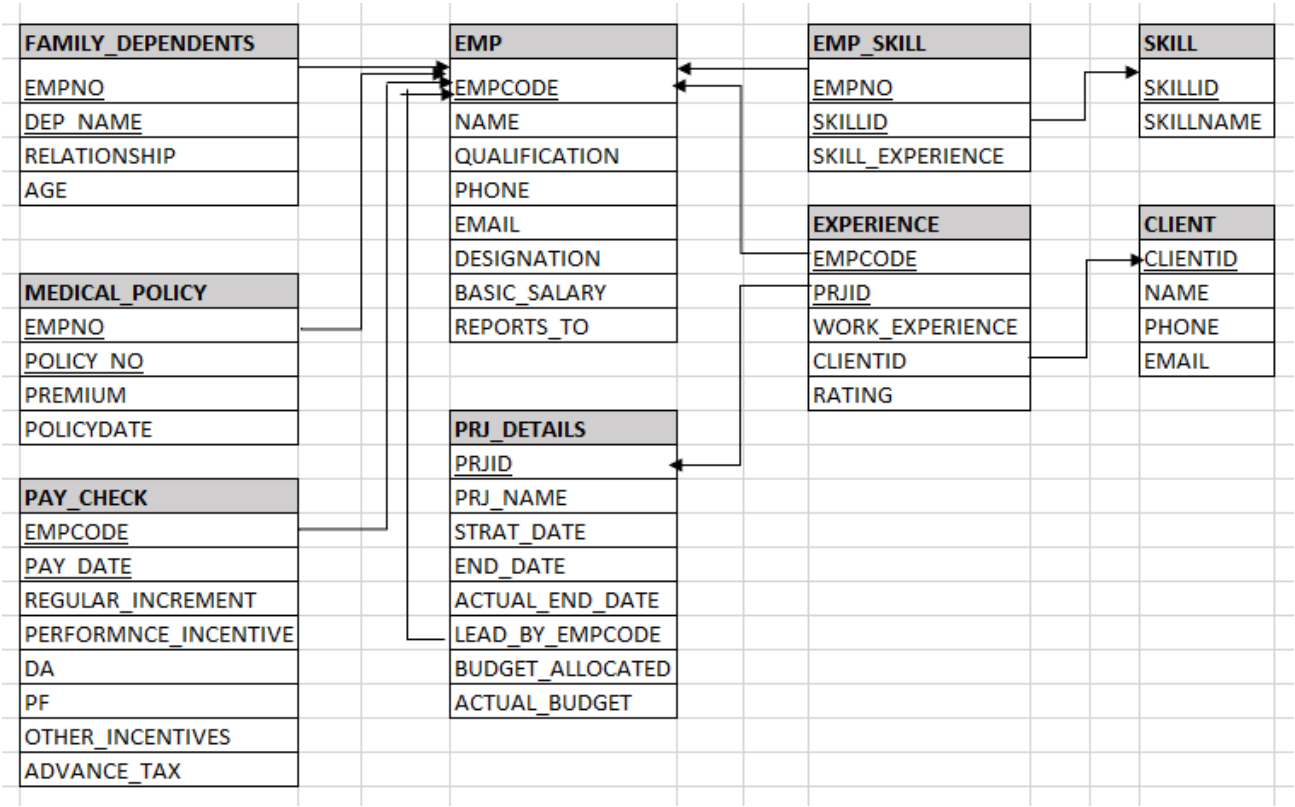
Week	Topics to be covered
1	SQL Basics – CREATE, ALTER, DROP
2	Populate and manipulate the database using INSERT, UPDATE, DELETE
3-5	Mini Project Abstract submission SQL Simple, join, subquery and Advanced Queries
6	Requirement specification submission PL/SQL
7	Cursors
8	Schema diagram and Schema implementation. Exception Handling
9	Triggers
10	Procedures, Functions, Packages
11	Mini Project Demo.
12	**End Term Practical Examination**

Course Outcomes (COs)

At the end of this course, the student should be able to:

		No. of Contact Hours
CO1:	Create and modify database objects	6
CO2:	Manipulate the data in the database	3
CO3:	Design queries to retrieve the data from database	9
CO4:	Perform database operations by integrating procedural language constructs	9
CO5:	Develop stored programs, application	6

Following is the schema diagram corresponding to database used in the lab manual.



Course Plan

No.	Topics	CO
L1	**1.0** Create the tables with the following columns and constraints with given constraint names:	CO1

EMP

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
EMPCODE	Number	3	Primary Key	PK_EMPNO
NAME	Varchar2	10		
QUALIFICATION	Varchar2	7	Valid Qualifications- BTECH, MTECH, MSC, MCA, PHD	VALID_QUALIF
PHONE	Number	10		
EMAIL	Varchar2	20	valid only if-contains **@digisol.com** as in email format	EMAIL_DOMAIN
DESIGNATION	Varchar2	20	Valid- DEVELOPER, PROJEACT LEADER, PROJECT MANAGER, TECHNICAL HEAD, PROJECT DIRECTOR, PRJOECT LEADER, CEO	VALID_DESIGNATION
SALARY	Number	7	97000 to 300000	SALARY_RANGE

SKILL

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
SKILLID	Char	3	Unique	UNQ_SKILLID
SKILLNAME	Varchar	20	Unique (not composite)	UNQ_SKILNAME

EMP_SKILL

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
EMPNO	Number	3	References EMP	FK_EMPNO
SKILLID	Char	3	References SKILL with **ON DELETE CASCADE**	FK_SKILLID
SKILL_EXPERIENCE	Number	3	>0 (in terms of Months)	

WORK_EXP

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
EMPCODE	Number	3	References EMP, Primary Key	Optional to give your own name
PRJID			References PRJ_DETAILS, Primary Key	
WORK_EXPERIENCE			>=0 NOT NULL	
CLIENTID			References CLIENT with **ON DELETE SET NULL**	
RATING			Valid values- A, B, C only	

CLIENT

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
CLIENTID	Char	3	Primary Key; start from 'C'	PK_CLIENTID; STRTSWITH_C
NAME	Varchar2	10		
PHONE	Char	14	Valid-if country code: +91-, +81-, + 31-, +48-, +44-; format +91-1234567890	PHONE_FORMAT

No.	Topics	CO

PRJ_DETAILS

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
PRJID	Char	3	Primary Key; start from 'P'	PK_PRJID
PRJ_NAME	Varchar2	10		
STRAT_DATE	Date			
END_DATE	Date		END_DATE > START_DATE	END_DATE_GARTER_START_DATE
ACTUAL_END_DATE			>=END_DATE	
LEAD_BY_EMPCODE	Number	3	References EMP	FK_LEAD_EMP
BUDGET_ALLOCATED	Number	10,1		
ACTUAL_BUDGET	Number	10,1		

FAMILY_DEPENDENTS

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
EMPNO			Primary Key; References EMP	
DEP_NAME	Varchar	20	Primary Key	
RELATIONSHIP	varchar	16		
AGE	Number	3	1 to 130	VALID_AGE

MEDICAL_POLICY

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
EMPNO			Primary Key; References EMP	Optional to give your own name
POLICYNO	Varchar2	6	Primary Key; starts from 'MED' e.g.: MED001	
PREMIUM	Number	5	>0	
POLICYDATE	Date			

PAY_CHECK

ATTRIBUTE	DATATYPE	SIZE	CONSTRAINTS	CONSTRAINTS NAME
EMPCODE			Primary Key; References EMP	
PAY_DATE	DATE		Primary Key	
REGULAR_INCREMENT	Number	4	2000/3000/5000/7000/9000	VALID_REGU_INCENTIVE
PERFORMNCE_INCENTIVE	Number	5		
DA	Number	7		
PF	Number	6		
OTHER_INCENTIVES	Number	4		
ADVANCE_TAX	Number	5		

1.1 Add column REPORTS_TO to EMP table with Foreign Key constraint refereing to EMPCODE and constraint name -FK_REPORTS_TO_EMPCODE.

1.2 Set a constraint on SKILLID column belonging to SKILL table- every column value starts with letter 'S'.

1.3 Add primary key constraint on EMPNO, SKILLID columns belonging to EMP_SKILL table having constraint name- PK_EMPNO_SKILLID.

1.4 Set Unique constraint on PRJ_NAME belonging to PRJ_DETAILS table with constraint name UNQ_PRJ_NAME.

No.	Topics	CO
	1.5 ADD column EMAIL to CLIENT & constraint with name 'VALID_EMAIL', valid only if-contains **@.** Symbols as in email format and there must be only 3 letters after**.** symbol on EMAIL column. **1.6** Add Unique Key constraint on PHONE belonging to EMP having constraint name -UNQ_PHONE. **1.7** Add Unique Key constraint on EMAIL belonging to EMP having constraint name -UNQ_EMAIL. **1.8** Drop primary key constraint from MEDICAL_POLICY table and add primary key constraint on EMPNO, POLICYNO, POLICYDATE. **1.9** Drop constraint from PREMIUM belonging to MEDICAL_POLICY table & add new constraint that PREMIUM amount must be more than 1000/-.	
L2	**Note:** Insert records into the following tables. • **Student can enter a valid value into the columns left blank, columns with NULL must be entered with NULL value only.** • **If some records cannot be inserted because of violation of constraints, then write the reason for violation in the record book and choose a data satisfying the constraint and insert them into the table. Write all such insert commands in the record book.** • **Insert 1 records of your own to each table and also write the corresponding insert commands in the record book.** **2.0 EMP table data.**	**CO2**

EMP

EMPCODE	NAME	QUALIFICATION	PHONE	EMAIL	DESIGNATION	SALARY	REPORTS_TO
100	RAJESH	BTECH		rajesh@digisol.com	DEVELOPER		101
101	RAVI	MTECH		ravi@digisol.com	PROJEACT LEADER		102
102	VIJAY	BTECH		vijay@digisol.com	PROJECT MANAGER		110
108	AJAY	BTECH		ajay@digisol.com	TECHNICAL HEAD		102
110	BHASKAR	MTECH		bhaskar@digisol.com	PROJECT DIRECTOR		106
106	RAJ	MSC		raj@digisol.com	PRJOECT LEADER		110
105	MANISH	MCA		manish@digisol.com	DEVELOPER		106
106	PRSAD	PHD		prasad@digisol.com	CEO		NULL
102	RAKESH	MTECH					
108	RAKESH	MTECH			Developer		
109		MBA					
110					S. ENGINEER		
111	RAJ			raj@digital.com			

No.	Topics	CO

2.1 SKILL table data.

SKILL

SKILLID	SKILLNAME
S1	Full stack
S2	Python
S3	Java
S4	C++
S6	Oracle
S7	Android
S8	AZURE
S9	DOT NET
S10	IOS

2.2 CLIENT table data

CLIENT

CLIENTID	NAME	PHONE	EMAIL
C1	ABC	+48-87489389	aaa@abc.com
C2	LTS	+91-78389397	bbb@lts.com
C3	RK	+44-67838934	ccc@rk.com
C4	MMA	+31-77839093	ddd@mma.com
C5	AKS	+81-66822903	eee@aks.com
C6	GKS	+78-75628982	fff@gks.com
C7	B&B	+91-64839839	ggg@bb.com
C8	ABT	+97-67554667	
C9	XYZ		xyz.ab.com

2.3 EMP_SKILL table data.

EMP_SKILL

EMPNO	SKILLID	SKILL_EXPERIENCE
100	S1	15
100	S3	10
101	S1	6
101	S2	10
102	S4	18
110	S1	14
110	S6	20
106	S2	10
106	S6	12
106	S8	10
108	S9	5
108	S6	18
105	S7	12
105	S10	11
101	S2	7

2.4 WORK_EXP table data

WORK_EXP

EMPCODE	PRJID	WORK_EXPERIENCE	CLIENTID	RATING
100	P1	20	C1	A
100	P3	15	C3	A
101	P1	16	C1	A
102	P2	18	C2	B
102	P1	18	C1	A
108	P2	12	C2	C
106	P3	20	C3	C
106	P4	10	C4	C
105	P2	19	C2	B
110	P4	12	C4	B
110	P5	16	C8	A
105	P5	5	C5	D
111	P5	5	C5	C

2.5 PRJ_DETAILS table data

PRJ_DETAILS

PRJID	PRJ_NAME	STRAT_DATE	END_DATE	ACTUAL_END_DATE	LEAD_BY_EMPCODE	BUDGET_ALLOCATED	ACTUAL_BUDGET
P1	SPYDER	10-01-2020	11-OCT-2022		102	2000000	1990000
P2	APOLLO	01-01-2019	10-JUL-2021	30-12-2021	108	3000000	3600000
P3	UNICORN	03-09-2021	10-OCT-2023		106	7500000	8000000
P4	ZEUS	28-02-2020	09-JUN-2021	10-01-2022	110	5500000	5000000
P5	JUPITOR	07-08-2018	04-FEB-2020	01-NOV-2020	110	9000000	9000000
P6	MARS	21-2-2022	07-AUG-2020		110		
NULL	XXX						
P7					123		

No.	Topics	CO

2.6 MEDICAL_POLICY table data

MEDICAL_POLICY

EMPNO	POLICYNO	PREMIUM	POLICYDATE
100	MED001	10000	10-12-2020
101	MED003	12000	15-12-2020
100	MED001	10000	10-01-2021
101	MED003	12000	15-01-2021
103	MED004	9000	21-01-2021

2.7 FAMILY_DEPENDENTS table data

FAMILY_DEPENDENTS

EMPNO	DEP_NAME	RELATIONSHIP	AGE
100	AJITH	SON	12
100	PRITHVI	DAGHTER	9
101	AARYA	DAGHTER	17
101	AKANKSH	SON	11
101	ANITHA	SPOUSE	43

2.8 PAY_CHECK (Leave Performance incentive, DA, PF columns blank (**NULL** values)

PAY_CHECK

EMPCODE	PAY_DATE	REGULAR_I	PERFORM	DA	PF	OTHER_IN	ADVANCE_TAX
100	01-12-2020	2000				2500	5000
101	01-12-2020	3000				5000	7000
102	01-12-2020	3000				3600	7000
100	01-01-2021	3000				450	4000
101	01-01-2021	5000				2100	4000
102	01-01-2021	5000				NULL	9000
103	02-01-2021	7000				900	3000

2.9 Run **COMMIT** command. Delete the employee having empcode 100 and confirm the result with justification.

2.10 Update email of employee with EMPCODE 100 to rajesh@digital.com and confirm the result with justification.

2.11 Update WORK_EXP by changing ProjId of employee 111 to P9 and confirm the result with reason.

2.12 Run **COMMIT** command. Delete the client information from CLIENT table having ClientID **C1**. Confirm the result with reason.

2.13 Run **COMMIT** command. Delete a family dependent information belonging to employee with employee id 100. Confirm the result. If records get deleted, run ROLLBACK to restore records back.

2.14 Update the EMP table to increase salary of DEVELOPERS by 1%.

2.15 Update employee code to which 108 Reports to as 106.

2.16 Update employee code to which 108 Reports to as 117. Confirm the result with reason. Type COMMIT;

No.	Topics	CO
L3	**3.1** Retrieve all Clients information **3.2** Display name, phone and email of employees having designation DEVELOPER. **3.3** Display name of employees who are working under employee with code 106. **3.4** Display name and email id of employees working as TECHNICAL HEAD and Designation BTECH. **3.5** Find name of employees who is drawing salary in the range 100000 to 200000. (two queries using comparison and BETWEEN) **3.6** List the Skills owned by the employee with Employee number 101. **3.7** Display the name of skills that employee with emp code 101 is having. **3.8** Retrieve Employee code of employees who worked with clients C1 or C2 and having rating either A or B. **3.9** Display the name of Employees who worked with client C2 and having Rating B. **3.10** List How many dependents an employee RAVI is having? **3.11** Retrieve employee name, Designation and Total annual salary (Salary*12) and rename the column as Annual_Salary. (using join) **3.12** List name of all employees and their dependents name if exist otherwise null. **3.13** List the project leader name and name of projects which have completed without time slippage and budget slippage. **3.14** Display employee name, Salary and pay check details corresponding to month -March. **3.15** Find the Employee name and his higher authority name, phone to who the employee reports. **3.16**. Display the project details in the descending order of their actual closing date. **3.17** Display Employee names, skill name in ascending order of employee number and descending order of their skill experience. **3.18** Display Employee name, dependent name, age where the relationship is SON and sort employee name in ascending and age in descending order. **3.19** Retrieve Employee names in Ascending order and Project names in descending order in which employees have worked. **3.20**. Find name of employees who have worked in SPYDER or APOLLO (use set Operator) **3.21** Find the name of employees who are having both skills Oracle & AZURE (use set Operator) **3.22** Retrieve the name of employees who do not work in project JUPITOR (use set Operator) **3.23** Display the name of employee who reports to none. **3.24** Display all employee names, email whose name starts with RA. **3.25** Retrieve the name of projects which are started between 01-01-2020 to 31-12-2020.	

No.	Topics	CO
	3.26 Display the name of employees who has at least few skills that employee with employee number 101 is having.	
L4	**4.1** List all employee names who are having same designation as that of employee 100. **4.2** Find the name of all employees who reports to Vijay. (use set membership) **4.3** List the name, designation and email of employees who have worked in the at least one of the projects in which Vijay has worked. (use set membership) **4.4** List the name of employees who has Daughters. **4.5** List the name of employees who do not work on SPYDER. (use set membership) **4.6** List the employees who have at least one skill that employees working on project P1 are possessing. (set comparison) **4.7** List the employee names who has more work experience in project P1 compared to experience of every employee working in project P2. (set comparison) **4.8** List the name of employees who worked in the project and client combinations in which Empcode 100 has worked. (Use set cardinality) **4.9** List the name of employees who didn't work on any of the project in which employee with empcode 106 has worked. (Use set cardinality) **4.10** Find the name of projects for which Allocated budget is at least more than the total salary of employees working on those projects. (correlation variable) **4.11** Display project name and the number of employees working on those projects and who obtained A or B ratings on the project they have worked. Sort the displayed records in descending order of number of employees worked. **4.12** Display the name of projects under which more than 2 employees are working. **4.13** Find Employee name and the number of dependents they are having. **4.14** Retrieve total premium amount collected each year. **4.15** Find employee name and having more than 20 months of total skill experience (irrespective of kind of skill) Use WITH clause. **4.16** Create a table called EMP_PRJ containing Employee name (Emp_Name), Project Budget (Budget) and Name of the project (Project_Name) which are led by employee with code 110. **4.17**. Increment the salary of employees by 10% for the employees who are PROJECT LEADER and 5% for employees who are DEVELOPER (use update ... case.) **4.18**. Create a view EMP_SKILL_EXPERIENCE containing employee name, skill_name & skill_experience where experience is more than 20 months. After creating view, update the skill_ experience of any employee and confirm the result with reasoning.	

No.	Topics	CO
	4.19 Create a view EMP_DESIGNATION containing Empno, Name, Salary of employees working with role DEVELOPER, PROJECT LEADER. After creating view, update the salary of any employee and confirm the result with reasoning. **4.20** Display employee name and number of projects in which they are working (using scalar sub query) **Optional**- Try writing 4.5 to 4.9 queries in alternate way.	
L5	**5.1** Display name of all name of employees in Upper, Lower, 1st letter capital format. **5.2** Display employee name and Designation of all the employees in the following format (using concatenation operator ||) "Rajesh Works as Developer." **5.3** Assume that you are displaying salary value in place with 10 space width. Display the salary so that remaining left side spaces after displaying salary value is padded with * characters. (LAPD). **5.4** Try the question 5.3 with RPAD. **5.5** Retrieve number of characters in employee names. **5.6** Retrieve substring part of employee email ids starting from 3rd position to 9th position characters. **5.7** Retrieve the emails of all employees by replacing digisol with digitech. **5.8** Retrieve employee name and Dependent-Name concatenated (concat ()) **5.9** Display ClientID and number of characters from Client table. **5.10** Retrieve right trimmed ClientID (RTRIM) and number of characters in the trimmed client id. **5.11** Try 5.10 with LTRIM and TRIM also. **5.12** Display ASCII value of 'A' and Character corresponding to ascii value 65. (ASCII, CHR) **5.13** Display the value 123.456 truncated to 2nd decimal position. **5.14** Consider any number (say 9) Retrieve square root, square and 5th power of the number. **5.15** Display the value 123.456 rounded to 2nd decimal position. **5.16** Retrieve absolute value of -123 and remainder after dividing 123 from 10. **5.17** Retrieve floor and ceiling value of 234.36 and -234.26. Floor(n)- largest integer value not greater than n and Ceil(n) - smallest integer value not less than n **5.18** Display start and end dates of projects in the format DD-Month-YYYY format. (use TO_CHAR) **5.19** Display month and Year on which projects are actually closed. (use TO_CHAR) **5.20** Display month, day, year extracted from Start_Date from Prj_Details. (use TO_CHAR) **5.21** Display the project name and the number of months between the start and end dates of the projects.	

No.	Topics	CO
	Use MONTHS_BETWEEN () **5.22** Display all the project name, end_date of the projects and the date after 2 months of end_date. Use ADD_MONTHS () **5.23** Find the date of next Thursday based on today's date.	
	WEEK-6 PL/SQL & CURSORS **6.1** Write PL/SQL block to display reverse of a number. **6.2** Write a PL/SQL block to check a given word is palindrome or not. **6.3** Write a PL/SQL block to count number of vowels, consonants, numbers and special characters in a sentence. **6.4** Write a PL/SQL block to accept an employee code from the user and display the employee details. **6.5** Write a PL/SQL block to accept an employee code and project id to find the rating obtained on the project. Depending on the rating assign an additional incentive (A-10000, B-5000, C- 3000). Display the information in the following format- **Employee Number:** **Employee Name:** **Salary:** **Project Name:** **Rating:** **Incentive:** **Total Salary:** Salary+(A-10000, B-5000, C- 3000) Update the Other_incentive of employee in Pay_check depending on rating. **6.6** Write a PL/SQL block to calculate Jan-2021 monthly salary of employee and other components of salary depending on Salary values in EMP table for the employee number 100. Display the salary information of employee 100 in the following format. **Name:** **Basic_Salary:** Salary from EMP table **Regular Increment:** Regular_Increment **Increased Basic Salary:** Salary+ Regular_Increment **Performance incentive:** performnce_incentive **DA:** 50% of Increased Basic Salary. **PF:** 12% of Increased Basic Salary. **Other Incentive:** other_incentives **Advance tax paid:** advance_tax **Monthly Gross Salary:** Salary+Regular_Increment+ Performnce_incentive+DA+PF+ other_incentives. **Monthly Take home salary:** Gross Salary – PF- advance_tax. **6.7** Do the salary processing of all employees according to the **Question 6**. (using **cursor**)	

No.	Topics	CO
	6.8 Create a Table ACCOUNT (Account_Number, Balance) with constraint on Balance that Balance must be more than 1000(minimum balance). Insert few records into the table. Write a PL/SQL block to accept and account number, withdrawal amount from the user. Update the Balance by deducting the withdrawal amount from the balance in the case updated balance is going to be more than minimum balance (1000) otherwise do not update the Balance. In both cases display proper message – Case 1: **Current Balance:** Balance Withdrawal of <*Withdrawal_amount*> is Successful Updated Current Balance: Updated balance after withdrawal Case 2: **Current Balance:** Balance Insufficient fund to withdraw, try with lesser withdrawal amount. **6.9** Write a PL/SQL block to display- Name of the project leader, Name of the projects handled for which budget is managed well within the Budget allocated. **6.10** Write a PL/SQL block to display the employees name and generate their email id in following format and display the same for all employees. **Name:** **Designation:** **Email Generated:** Name.Designation@digisoul.com **6.11** Write the PL/SQL block for Question 8. along with exception handling for- i. Entry of not existing Account number. ii. If Resulting balance is lesser than minimum Balance. **6.12** Assume that we have a new project *Marvel* and there is need for human resources with skills- 3 full stack developer, 1 Oracle and 1 Azure each having at least 12 months of experience. Pass skill_experience as parameter and list all the human resources available in the company satisfying the need. **6.13** Write PL/SQL block with parameterized cursor to display all the employee code, name, designation of the employees who are rated with a particular rating by a particular client. Pass rating and client as parameter to the cursor. (Use cursor for loop.) **6.14** Write a PL/SQL block to display name, total work experience of 5 employees who are having highest total work experience.	
	WEEK-7 STORED PROCEDURE, FUNCTION & PACKAGES **7.1** Create a Table ACCOUNT (Account_Number, Balance) with constraint on Balance that Balance must be more than 1000(minimum balance). Insert few records into the table. Write a stored procedure to accept and account number, withdrawal amount from the user. Update the Balance by deducting the withdrawal amount from the balance in the case updated balance is going to be more than minimum balance (1000) otherwise do not update the Balance. In both cases display proper message – **Case 1**: **Current Balance:** Balance	

No.	Topics	CO
	Withdrawal of <*Withdrawal_amount*> is Successful Updated Current Balance: Updated balance after withdrawal **Case 2:** **Current Balance:** Balance Insufficient fund to withdraw, try with lesser withdrawal amount. Use PL/SQL block to call the stored procedure. **7.2** Assume that we have a new project *Marvel* and there is need for human resources with certain skill having at least 12 months of experience. Write a stored procedure to pass skill, skill_experience as parameter and list all the human resources available in the company satisfying the need. Create a PL/SQL block to call procedure. **7.3** Write a stored procedure to take project id as input parameter to find additional budget needed depending on how many days additionally project is executed (depending on actual_end_Date) Additional Budget Needed= Additional days * per day cost of project per day cost of project = (Budget Allocated)**/** (Duration of project in days (depending on End_date & Start_Date)) Return Additional Budget Needed back to calling PL/SQL block. The calling PL/SQL block must display information in the following format- Name of Project Leader: Name & email of Employee to whom Project leader reports to: Additional Budget Needed: **7.4** Write a stored function to accept project id of a project and return an incentive to be given to all employee working on the project depending on the rating obtained by them. In calling PL/SQL allow the user to enter project id to be passed to the function. A-30000; B- 20000; C- 10000 **7.5** Write a stored function to accept phone a country code (such as +91, +48 etc.) to find the name of clients in those countries. **7.6** Write a stored function to generate email ids of faculty automatically using name, empno as-name_empno@digisol.com. **7.7** Create a package containing following procedure & functions- i. A function to accept employee number to retrieve employee name to whom the employee reports. ii. A stored procedure to accept Project id and to display name of employees working on the project. iii. Write a stored procedure display project name started from a date input by the user. iv. Write a stored procedure to display details of work force available (such as Empno, Name, Skill Experience) for each skill in the Skill table. v. Write a function to find number of employees with each designation.	

No.	Topics	CO
	WEEK-8 TRIGGERS **8.1** Write a trigger to be fired when basic salary hike is given to employees. Hike range and range of basic salary for each kind of designation is as below-	

Designation	Regular_Increment RANGE	Basic_salary RANGE
DEVELOPER	1000	97000 - 120000
PROJECT LEADER	2000	110000-140000
PROJECT MANAGER	3000	120000-160000
TECHNICAL HEAD	3000	140000-200000
PROJECT MANAGER	5000	180000-250000
PROJECT DIRECTOR	10000	210000-270000
CEO	20000	220000-300000

Whenever basic salary is updated, depending on the designation of the employee the trigger should check the increment range is legal and increased basic salary is in the range of valid basic salary corresponding to the designation. If anyone of these conditions is violated then cancel the update operation.

8.2 Design a trigger to be fired when a new project detail is added or Lead_by_Empcode is updated. The trigger should check the Lead_by_Empcode is valid as per company policy. The company policy says-an employee can lead a project only if he has designation – Project Lead or Project Manager and the total skill experience is 18 or more than 18 months. If these conditions are not met then cancel the insert/update operation on project details table.

8.3 Develop a trigger to monitor the modification of Actual end dates and actual budgets of projects. The trigger should allow only if time extension is less than 90 days and additional budget is not more than 15% of actual budget allocated. Violation of any of these rules must flash error message – "*Need to take Approval through proper channel*".

8.4 Write a trigger to be fired when an obsolete skill record is deleted from SKILL table. If only one employee with that skill exists then don't delete from EMP_SKILL otherwise corresponding employees with those obsolete skills must be removed from EMP_SKILL except one employee with highest skill experience. In a main PL/SQL block ask the user to input skill id which is obsolete and to be removed and execute delete statement corresponding to that skidlid.

L9	**Mini Project Implementation**	
L10	**Mini Project Demo**	
L11	**END SEMESTER LAB EXAM**	

References

1. Ivan Bayross, "SQL, PL/SQL-The Programming Language of ORACLE", 4th Edition, BPB Publications,

2. Satish Asnani, “Oracle Database 11g”, PHI, 2010.
3. Scott Urman, “ORACLE – PL/SQL Programming”, Oracle Press.

Submitted by: Vinayak Mantoor & Archana.H

(Signature of the faculty)

Date: 12/03/2022

Approved by: Dr. Karunakar A.K

(Signature of HOD)

Date: 12/03/2022