

DBMS LAB NAME : _____ Rollno : _____ Section : _____

Q.1. Get all the data of all jobs.

SQL> *Select * from J;*

OUTPUT:

JNO	JNAME	CITY
J1	Salter	Paris
J2	Punch	Rome
J3	Header	Athens
J4	Congole	Athens
J5	Callisto	London
J6	Tenma	Oslo
J7	Pina	Londn

1 row selected.

Q.2. Get all the data on all jobs in London.

SQL> *Select * from J where city = 'LONDON';*

OUTPUT:

JNO	JNAME	CITY
J5	Callisto	London
J7	Pina	London

Q.3. Get supplier numbers for suppliers (S table) who supply (SPJ table) job J1, in supplier number order.

SQL> *Select distinct sno from spj where snum = J1 order by sno;*

OUTPUT:

100
101
102
103

Q.4. Get all shipments (SPJ table) where the quantity is in the range 300 to 750 inclusive.

SQL> *Select * from spj where qty between 300 and 750;*

*Select sal as sal0 as PNO, JNO as JNO, qty from SPJ where qty >= 300
and qty <= 750;*

Q.1. Get all the data of all jobs.

SQL > *Select * from J;*

OUTPUT:

JNO	JNAME	CITY
J1	Sorter	Paris
J2	Punch	Rome
J3	Reader	Athens
J4	Console	Athens
J5	Collator	London
J6	Terminal	Oslo
J7	Tape	London

7 rows selected.

Q.2. Get all the data on all jobs in London.

SQL > *Select * from J where city = 'LONDON';*

OUTPUT:

JNO	JNAME	CITY
J5	Collator	London
J7	Tape	London

Q.3. Get supplier numbers for suppliers (S table) who supply (SPJ table) job J1, in supplier number order.

SQL > *Select distinct sno from spj where sn=j1' order by sno;*

OUTPUT:

SNO
S1
S2
S3

Q.4. Get all shipments (SPJ table) where the quantity is in the range 300 to 750 inclusive.

SQL > *Select * from spj where qty between 300 and 750;*

*Select sno as sno, JN as JNO, qty from SPJ where qty >= 300
and qty <= 750;*

OUTPUT:

SNO	PNO	JNO	QTY
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J4	500
S2	P3	J5	600
S2	P3	J6	400
S3	P4	J2	500
S4	P6	J3	300
S4	P6	J7	300
S5	P5	J5	500
S5	P5	J4	400
S5	P6	J4	500

11 rows selected.

Q.5. For each supplier, tell the total number of parts supplied to some project.

SQL> Select sno, sum(qty) from spj group by sno; ~~order by sno;~~**OUTPUT:**

SNO	SUM (QTY)
S1	900
S2	3200
S3	700
S4	600
S5	3100

Q.6. Get a list of all part-color, part-city combinations, with duplicate (color, city) eliminated.

SQL> Select distinct ~~sno, pno, jno~~ from spj where sno <= 5 order by sno;
color, city from p order by city;**OUTPUT:**

COLOR	CITY
Red	London
Blue	Paris
Green	Paris
Blue	Rome

Q.7. Get the total number of jobs supplied by supplier S1.

SQL> Select * from spj where sno between 200 and 700;
Select count(jno) from spj where sno = 'S1';

DBMS LAB

NAME :

Rollno :

Section :

OUTPUT:

COUNT (DISTINCT ING)

2

Q.8. Get the total quantity of part P1 supplied by supplier S1.

SQL > *Select sum(qty) from SPJ where sno = 'S1' and pno = 'P1';*

OUTPUT:

SUM (QTY)

300

Q.9. For each part being supplied to a job, get the part number, the job number and the corresponding total quantity.

SQL > *Select Jno, Pno, sum(qty) from SPJ group by (Jno, Pno);*

OUTPUT:

JNO PNO SUM (QTY)

J1 P1 300

J1 P2 100

J1 P3 100

J1 P4 100

J1 P5 100

J1 P6 100

J1 P7 100

J1 P8 100

J1 P9 100

J1 P10 100

J1 P11 100

J1 P12 100

J1 P13 100

J1 P14 100

J1 P15 100

J1 P16 100

J1 P17 100

J1 P18 100

J1 P19 100

J1 P20 100

J1 P21 100

J1 P22 100

J1 P23 100

J1 P24 100

J1 P25 100

J1 P26 100

J1 P27 100

J1 P28 100

J1 P29 100

J1 P30 100

J1 P31 100

J1 P32 100

J1 P33 100

J1 P34 100

J1 P35 100

J1 P36 100

J1 P37 100

J1 P38 100

J1 P39 100

J1 P40 100

J1 P41 100

J1 P42 100

J1 P43 100

J1 P44 100

J1 P45 100

J1 P46 100

J1 P47 100

J1 P48 100

J1 P49 100

J1 P50 100

J1 P51 100

J1 P52 100

J1 P53 100

J1 P54 100

J1 P55 100

J1 P56 100

J1 P57 100

J1 P58 100

J1 P59 100

J1 P60 100

J1 P61 100

J1 P62 100

J1 P63 100

J1 P64 100

J1 P65 100

J1 P66 100

J1 P67 100

J1 P68 100

J1 P69 100

J1 P70 100

J1 P71 100

J1 P72 100

J1 P73 100

J1 P74 100

J1 P75 100

J1 P76 100

J1 P77 100

J1 P78 100

J1 P79 100

J1 P80 100

J1 P81 100

J1 P82 100

J1 P83 100

J1 P84 100

J1 P85 100

J1 P86 100

J1 P87 100

J1 P88 100

J1 P89 100

J1 P90 100

J1 P91 100

J1 P92 100

J1 P93 100

J1 P94 100

J1 P95 100

J1 P96 100

J1 P97 100

J1 P98 100

J1 P99 100

J1 P100 100

J1 P101 100

J1 P102 100

J1 P103 100

J1 P104 100

J1 P105 100

J1 P106 100

J1 P107 100

J1 P108 100

J1 P109 100

J1 P110 100

J1 P111 100

J1 P112 100

J1 P113 100

J1 P114 100

J1 P115 100

J1 P116 100

J1 P117 100

J1 P118 100

J1 P119 100

J1 P120 100

J1 P121 100

J1 P122 100

J1 P123 100

J1 P124 100

J1 P125 100

J1 P126 100

J1 P127 100

J1 P128 100

J1 P129 100

J1 P130 100

J1 P131 100

J1 P132 100

J1 P133 100

J1 P134 100

J1 P135 100

J1 P136 100

J1 P137 100

J1 P138 100

J1 P139 100

J1 P140 100

J1 P141 100

J1 P142 100

J1 P143 100

J1 P144 100

J1 P145 100

J1 P146 100

J1 P147 100

J1 P148 100

J1 P149 100

J1 P150 100

J1 P151 100

J1 P152 100

J1 P153 100

J1 P154 100

J1 P155 100

J1 P156 100

J1 P157 100

J1 P158 100

J1 P159 100

J1 P160 100

J1 P161 100

J1 P162 100

J1 P163 100

J1 P164 100

J1 P165 100

J1 P166 100

J1 P167 100

J1 P168 100

J1 P169 100

J1 P170 100

J1 P171 100

J1 P172 100

J1 P173 100

J1 P174 100

J1 P175 100

J1 P176 100

J1 P177 100

J1 P178 100

J1 P179 100

J1 P180 100

J1 P181 100

J1 P182 100

J1 P183 100

J1 P184 100

J1 P185 100

J1 P186 100

J1 P187 100

J1 P188 100

J1 P189 100

J1 P190 100

J1 P191 100

J1 P192 100

J1 P193 100

J1 P194 100

J1 P195 100

J1 P196 100

J1 P197 100

J1 P198 100

J1 P199 100

J1 P200 100

J1 P201 100

J1 P202 100

J1 P203 100

J1 P204 100

J1 P205 100

J1 P206 100

J1 P207 100

J1 P208 100

J1 P209 100

J1 P210 100

J1 P211 100

J1 P212 100

J1 P213 100

J1 P214 100

J1 P215 100

J1 P216 100

J1 P217 100

J1 P218 100

J1 P219 100

J1 P220 100

J1 P221 100

J1 P222 100

J1 P223 100

J1 P224 100

J1 P225 100

J1 P226 100

J1 P227 100

J1 P228 100

J1 P229 100

J1 P230 100

J1 P231 100

J1 P232 100

J1 P233 100

J1 P234 100

J1 P235 100

J1 P236 100

J1 P237 100

J1 P238 100

J1 P239 100

J1 P240 100

J1 P241 100

J1 P242 100

J1 P243 100

J1 P244 100

J1 P245 100

J1 P246 100

J1 P247 100

J1 P248 100

J1 P249 100

J1 P250 100

J1 P251 100

<p

Q.10. Get part numbers for parts supplied to some job in the average quantity of more than 320.

SQL > *Select pno from SPJ group by pno having avg (Qty) > 320;*

OUTPUT :

PNO

P4
P5
P6
P7
P8

Q.11. Get all shipments where the quantity is non-null.

SQL > *Select * from SPJ where qty is not NULL;*

OUTPUT :

ENO	PNO	JNO	QTY
E1	P1	J1	200
E1	P1	J4	700
E2	P3	J1	400
E2	P3	J2	200
E2	P3	J3	200
E2	P3	J4	500
E2	P3	J5	600
E2	P3	J6	400
E2	P3	J7	800
E3	P3	J2	100
E3	P3	J1	200
E3	P4	J2	500
E4	P6	J3	300
E4	P6	J7	300
E5	P2	J2	200
E5	P2	J4	100
E5	P4	J1	500
E5	P5	J2	100
E5	P5	J7	200
E6	P6	J2	200
E6	P7	J4	100
E6	P4	J6	200
E6	P7	J4	200
E7	P1	J4	400
E7	P5	J7	500

34 rows selected.

Q.12. Get project numbers and cities where the city has an "o" as the second letter of its name.

SQL> Select Jno, city from J where city like ' _o %';

OUTPUT:

JNO CITY

J2 Rome
J5 London
J7 London

Q.13. Get supplier names which start with the letters 'Sm'.

SQL> Select name from S where name like 'Sm%';

OUTPUT:

SNAME

Smith

Q.14. Get supplier names that have a letter 'e' somewhere in their name.

SQL> Select name from S where name like '%e%';

OUTPUT:

SNAME

Jones
Blake

Q.15. Get the part number and total shipment quantity for each part.

SQL> Select Pno , sum(qty) from SPJ group by Pno;

OUTPUT:

PNO SUM(QTY)

P4 1300
P1 1000

P2	300
P3	3500
P6	1300
P5	1100

6 rows selected.

Q.16. Get the last five shipments.

SQL> Select * from SPJ minus select * from SPJ where rownum <= 5
 Select count(*) - 5 from SPJ;

OUTPUT:

SNO	PNO	JNO	QTY
S5	P6	J4	500
S3	P5	J4	400
S2	P4	J4	800
S3	P3	J4	200
S5	P1	J4	100

Q.17. Get the first five shipments.

SQL> Select * from SPJ where rownum <= 5;

OUTPUT:

SNO	PNO	JNO	QTY
S1	P1	J1	200
S1	P1	J4	700
S2	P3	J1	400
S2	P3	J2	200
S2	P3	J3	200

Q.18. Get part numbers for all parts supplied by more than two supplier

SQL> Select distinct d.pno from SPJ a where 2 < (Select count(sno) from SPJ where Pno = a.pno)

OUTPUT:

P2

P1
P3
P6
P5

DBMS LAB

NAME: _____

Rollno:

Section :

Q.19. Get part numbers for parts supplied to some project in Paris.

SOL: Select distinct pin from SPJ natural join j where city = 'Paris';

OUTPUT:

Q.20. Get part numbers for parts that are not supplied to any project in Paris.

SOL: select distinct pro from SPJ where minisect distinct pro from SPJ natural join where city = 'Paris';

OUTPUT:

Q.21. Get all supplier-number, part-number, job-number triples such that the indicated supplier, part, job are in the same city.

SOL - Select sno,pho , Inc from S,P,J where S.city = P.city and
P.city = J.city;

OUTPUT:

Q.22. Get part number for parts supplied (SPJ table) by a supplier in London.

SQL > Select distinct pno from SPJ natural join S where city = 'London';

OUTPUT:

PNO

P1

P2

P3

P4

P5

P6

P7

P8

P9

P10

P11

P12

P13

P14

P15

P16

P17

P18

P19

P20

P21

P22

P23

P24

P25

P26

P27

P28

P29

P30

P31

P32

P33

P34

P35

P36

P37

P38

P39

P40

P41

P42

P43

P44

P45

P46

P47

P48

P49

P50

P51

P52

P53

P54

P55

P56

P57

P58

P59

P60

P61

P62

P63

P64

P65

P66

P67

P68

P69

P70

P71

P72

P73

P74

P75

P76

P77

P78

P79

P80

P81

P82

P83

P84

P85

P86

P87

P88

P89

P90

P91

P92

P93

P94

P95

P96

P97

P98

P99

P100

P101

P102

P103

P104

P105

P106

P107

P108

P109

P110

P111

P112

P113

P114

P115

P116

P117

P118

P119

P120

P121

P122

P123

P124

P125

P126

P127

P128

P129

P130

P131

P132

P133

P134

P135

P136

P137

P138

P139

P140

P141

P142

P143

P144

P145

P146

P147

P148

P149

P150

P151

P152

P153

P154

P155

P156

P157

P158

P159

P160

P161

P162

P163

P164

P165

P166

P167

P168

P169

P170

P171

P172

P173

P174

P175

P176

P177

P178

P179

P180

P181

P182

P183

P184

P185

P186

P187

P188

P189

P190

P191

P192

P193

P194

P195

P196

P197

P198

P199

P200

P201

P202

P203

P204

P205

P206

P207

P208

P209

P210

P211

P212

P213

P214

P215

P216

P217

P218

P219

P220

P221

P222

P223

P224

P225

P226

P227

P228

P229

P230

P231

P232

P233

P234

P235

P236

P237

P238

P239

P240

P241

P242

P243

P244

P245

P246

P247

P248

P249

P250

P251

P252

P253

P254

P255

P256

P257

P258

P259

P260

P261

P262

P263

P264

P265

P266

P267

P268

P269

P270

P271

P272

P273

P274

P275

P276

P277

P278

P279

P280

P281

P282

P283

P284

P285

P286

P287

P288

P289

P290

P291

P292</p

Q.25. Get part numbers for parts supplied to any job by a supplier in the same city as the job.

SQL > *Select pno from spj, s, j where spj.sno = s.sno and spj.jno = j.jno
and s.city = j.city;*

OUTPUT:

```
PNO
-----
P3
P3
P6
P1
P2
P3
P4
P5
P6
```

9 rows selected.

Q.26. Get job numbers for jobs supplied by at least one supplier not in the same city.

SQL > *Select unique jno from spj, s, p where spj.sno = s.sno and spj.pno =
p.pno and s.city ≠ p.city*

OUTPUT:

```
JNO
-----
J2
J3
J4
J5
J6
J7
J8
```

7 rows selected.

Q.27. Get all pairs of part numbers such that the same supplier supplies both the indicated parts.

SQL > *Select distinct a.pno = b.pno from spj a,spj b where a.sno = b.sno
and a.pno < b.pno;*

OUTPUT:

PNO	PNO
P3	P2
P4	P5
P5	P6
P2	P4
P4	P5
P3	P3
P4	P4
P2	P6
P3	P3
P5	P5
P6	P6
P4	P4
P3	P3
P5	P5
P6	P6
P4	P4
P3	P3
P5	P5
P6	P6

15 rows selected.

Q.28. Get job names for jobs supplied by supplier S1.

SQL> Select name as jname from j, spj where j.jno = spj.jno and
 spj.sno = 'S1';

OUTPUT:

NAME
Shutter
Console

Q.29. Get colors for parts supplied by supplier S1.

SQL> Select unique color from p, spj where p.pno = spj.pno and
 spj.sno = 'S1';

OUTPUT:

COLOR
Blue

Q.30. Get part numbers for parts supplied to any job in London.

SQL> Select unique pno from spj, j where spj.jno = j.jno and j.city
 = 'London';

DBMS LAB

NAME :

Rollno :

Section :

OUTPUT:

PNO

P3

P6

P5

Q.31. Get job numbers for jobs using at least one part available from supplier S1.

SQL> Select distinct jno from spj where pno in (select pno from spj where sno = 'S1'));

OUTPUT:

JNC

J1

J4

Q.32. Get supplier numbers for suppliers supplying at least one part supplied by at least one supplier who supplies at least one red part.

SQL> Select distinct sno from spj where pno in (select distinct pno from spj where sno in (select distinct spj.sno from spj, p where p.color = 'Red' and spj.pno = p.pno));

OUTPUT:

SNO

S3

S4

S5

S2

S1

Q.33. Get supplier numbers for suppliers with status lower than that of supplier S1.

SQL> Select sno from s where st < (select status from supplier where sno = 'S1'));

OUTPUT:

SNO

S2

S3

S4

S5

S6

S7

S8

S9

S10

S11

S12

S13

S14

S15

S16

S17

S18

S19

S20

S21

S22

S23

S24

S25

S26

S27

S28

S29

S30

S31

S32

S33

S34

S35

S36

S37

S38

S39

S40

S41

S42

S43

S44

S45

S46

S47

S48

S49

S50

S51

S52

S53

S54

S55

S56

S57

S58

S59

S60

S61

S62

S63

S64

S65

S66

S67

S68

S69

S70

S71

S72

S73

S74

S75

S76

S77

S78

S79

S80

S81

S82

S83

S84

S85

S86

S87

S88

S89

S90

S91

S92

S93

S94

S95

S96

S97

S98

S99

S100

S101

S102

S103

S104

S105

S106

S107

S108

S109

S110

S111

S112

S113

S114

S115

S116

S117

S118

S119

S120

S121

S122

S123

S124

S125

S126

S127

S128

S129

S130

S131

S132

S133

S134

S135

S136

S137

S138

S139

S140

S141

S142

S143

S144

S145

S146

S147

S148

S149

S150

S151

S152

S153

S154

S155

S156

S157

S158

S159

S160

S161

S162

S163

S164

S165

S166

S167

S168

S169

S170

S171

S172

S173

S174

S175

S176

S177

S178

S179

S180

S181

S182

S183

S184

S185

S186

S187

S188

S189

S190

S191

S192

S193

S194

S195

S196

S197

S198

S199

S200

S201

S202

S203

S204

S205

S206

S207

S208

S209

S210

S211

S212

S213

S214

S215

S216

S217

S218

S219

S220

S221

S222

S223

S224

S225

S226

S227

S228

S229

S230

S231

S232

S233

S234

S235

S236

S237

S238

S239

S240

S241

S242

S243

S244

S245

S246

S247

S248

S249

S250

S251

S252

S253

S254

S255

S256

S257

S258

S259

S260

S261

S262

S263

S264

S265

S266

S267

S268

S269

S270

S271

S272

S273

S274

S275

S276

S277

S278

S279

Q.34. Get job numbers for jobs whose city is first in the alphabetic list of such cities.

SQL > `Select jno from j where city = (select min(city) from j);`

OUTPUT:

101

102

Q.35. Get job numbers for jobs supplied with part P1 in an average quantity greater than the greatest quantity in which any part is supplied to project J3.

SQL > `Select jno from spj where pno='P1' group by jno having avg(qty) > (select max(avg(qty)) from spj where jno='J3');`

OUTPUT:

101

102

Q.36. Get supplier numbers for suppliers supplying some job with part P1 in a quantity greater than the average shipment quantity of part P1 for that project.

SQL > `Select sno from spj A where A.pno = 'P1' and qty > (select avg(qty) from spj B where B.pno = 'P1' and A.pno = B.pno);`

OUTPUT:

101

102

Q.37. Get part numbers from parts supplied to any job in London.

SQL > `Select unique pno from spj, j where spj.jno=j.jno and j.city = 'London';`

OUTPUT:

101

102

Q.38. Get job numbers for jobs using at least one part available from suppliers S1 i.e, we know that S1 shipped that part.

SQL > Select distinct jno from spj where pno in (select pno from spj where sno = 'S1');

OUTPUT :

JNO

J1

J4

Q.39. Get job numbers for jobs not supplied with any red part by any London supplier.

SQL > Select jno from spj minus select jno from spj, s, p where spj.sno = s.sno and spj.pno = p.no and p.color = 'Red' and s.city = 'London';

OUTPUT :

JNO

J2

J5

J6

Q.40. Get job numbers for jobs supplied entirely by supplier S2.

SQL > Select jno from j where not exists (select * from spj where j.jno = spj.jno and s.no = 'S2');

OUTPUT :

JNO

J6

Q.41. Get part numbers for parts supplied to all jobs in London.

SQL > Select pno from p where not exists (select * from j where city = 'London' and not exists (select * from spj where p.pno = spj.pno and j.jno = spj.jno));

OUTPUT :

PNO

P1

P2

DBMS LAB NAME :

Rollno : Section :

Q.42. Construct a list of all cities in which at least one supplier, part, or job is located.

SQL > *Select city from s union select city from p union select city from j;*

OUTPUT:

CITY

Athens
London
Paris
Salzburg
Berlin

Q.43. Get the colors of parts either whose city is London, or Paris or both.

SQL > *Select distinct color from p where city = 'London' or city = 'Paris'
group by color;*

OUTPUT:

COLOR

Blue
Green
Red

Q.44. How many suppliers in Athens that supply red parts?

SQL > *Select count (spj.sno) from spj, p, s where spj.pno = p.pno and spj.sno =
s.sno and p.color = 'Red' and s.city = 'Athens';*

OUTPUT:

COUNT (SPJ.SNO)

4

Q.45. Find sname of suppliers who do not supply any part heavier than 18 (to any project).

SQL > *Select sname as Sname from s where sno not in (select sno from
spj natural join p where weight > 18) order by 1;*

OUTPUT:

SNAME

Blake
Jones
Smith

DBMS LAB NAME :

Rollno : Section :

Q.46. Find number and name of suppliers that supplies a 'Nut'.

SQL > Select distinct (name) from s, spj where s.sno = spj.sno and spj.pno = 'P1';

OUTPUT:

EMMA

Adams
Smith

Q.47. Find number of projects that do not use any locally made parts (i.e., if the project takes place in city x, then it does not use any part made in city x).

SQL > Select distinct s1.pno from spj, s1 natural join j where j.city > (select city from p where pno >= s1.pno) minus select distinct s1.pno from spj natural join j where j.city = (select city from p where pno >= s1.pno);

OUTPUT:

PNO

P1
P2
P4
P5

Q.48. Get suppliers for whom the total shipment quantity, taken over all shipments for the supplier is less than 1000.

SQL > Select * from s natural join (select sno from spj group by sno having sum(qty) < 1000) Order by sno;

OUTPUT:

SNO	CITY
S1	London
S4	London
S3	Paris

STATUS CITY

20	London
20	London
30	Paris

Q.49. Get suppliers for whom the minimum shipment quantity is less than half the maximum shipment quantity (taken overall shipments for the supplier in both cases).

SQL > Select * from s natural join (select sno from spj group by sno having min(qty) < 0.5 * max(qty)) group by sno;

OUTPUT:

SNO	SNAME	STATUS CITY
S1	Smith	20 London
S2	Jones	10 Paris
S3	Blake	30 Paris
S4	Adams	30 Athens

Q.50. For each supplier, find supplier details and total, maximum, and minimum shipment quantity, taken over all shipments for the supplier.

SQL > *Select * from S natural join (select sno, sum(qty) "TOTQ", max(qty) "MAXQ", min(qty) "MINQ" from SPJ group by sno) order by status;*

OUTPUT:

SNO	SNAME	STATUS CITY	TOTQ	MAXQ	MINQ
S2	Jones	10 Paris	3200	800	100
S1	Smith	20 London	900	700	200
S4	Clark	20 London	600	300	300
S5	Adams	30 Athens	3100	800	500
S3	Blake	30 Paris	700	500	200

(B)
Rohit
Teacher I/C
Prof. Dinesh Kumar Bhawnani

The use of a sequence can be seen in the following example.

Example

Insert a new employee 'BILL' with a designation 'CLERK' and salary of 1000. He is to be posted to the department 20.

This can be done using the following query.

Query

```
SQL> INSERT INTO emp (empno, ename, job, sal, dept)
      VALUES (empseq.nextval, 'BILL', 'CLERK', 1000, 20);
```

Deleting a Sequence

To delete a sequence from the database use DROP SEQUENCE command

Syntax

```
DROP SEQUENCE [schema.]sequence_name
```

Where

- Schema – It is the schema containing the sequence. The default assumes the sequence is in our own schema.
- Sequence_name – The name of the sequence to be dropped.

For example, to drop the sequence empseq use the following command

```
SQL> DROP SEQUENCE empseq;
```

Q.1. Create a view emp30 which display the name and job titles of employee working in department 30.

SQL> ~~Create view~~ Create view emp30 as select ename, job from emp
where deptno=30;

OUTPUT:

```
SQL> select * from emp30;
```

ENAME	JOB
ALLEN	SALARIAN
WARD	SALESMAN
MARTIN	SALESMAN
STANT	MANAGER
TURNER	SALESMAN
JAMES	CLERK

6 rows selected.

Q.2. Create a view empcount which counts the number of employees working in each department.

SQL > Create view empcount as select deptno, count(emplno) as noofemp
from emp group by deptno;

OUTPUT:

SQL> select * from empcount;

DEPTNO	NOOFEPL
10	7
20	3
30	4
40	1

Q.3. Create a view empsales which display the name and job titles of employee working in department 'SALES'.

SQL > Create view empsales as select ename, job from emp, dept
where ~~deptno~~ emp.deptno = deptno and dname = 'SALES';

OUTPUT:

SQL> select * from empsales;

ENAME	JOB
ALLEN	SALESMAN
WARD	SALESMAN
MARTIN	SALESMAN
BLAKE	MANAGER
TURNER	SALESMAN
JAMES	CLERK

6 rows selected.

Q.4. Create a view emploc which display the name and job of employees working in location 'CHICAGO'.

SQL > Create view emploc as select ename, job from emp, dept
where emp.deptno = deptno and loc = 'CHICAGO';

DBMS LAB

NAME :

Rollno :

Section :

OUTPUT:

SQL> select * from emploc;

ENAME	JOB
ALLEN	SALESMAN
WARD	SALESMAN
MARTIN	SALESMAN
BLAKE	MANAGER
TURNER	SALESMAN
JAMES	CLERK

6 rows selected.

- Q.5. Create a view emptotalsal which find the name and total salary employees department wise and display only those departments which have at least 5 employees.

SQL> Create view emptotalsal as select deptno, sum(sal), as totalsal
from emp natural join dept having count(db distinct ename) >= 5
group by deptno;

OUTPUT:

SQL> select * from emptotalsal;

DEPTNO	TOTALSAL
30	9400
20	10875

APP

Dinesh Kumar Bhawnani
Teacher I/C

Prof. Dinesh Kumar Bhawnani

Lab 8 Introduction to PL/SQL

Q.1. Write a PL/SQL program to input a name and display it.

PL/SQL Program :

```
declare
    name varchar(20);
begin
    declare cursor for select name from dual;
    for i in cursor loop
        dbms_output.put_line('Your name is '||i.name);
    end loop;
end;
```

OUTPUT:

```
Enter value for name: dinesh
old 4: name := 'name';
new 4: name := 'dinesh';
Your name is dinesh
```

PL/SQL procedure successfully completed.

Q.2 Write a PL/SQL program to input 2 numbers and display addition, subtraction, multiplication, division and modulus of these 2 numbers.

PL/SQL Program

```

declare
    a Number := 'f a';
    b Number := 'f b';

begin
    dbms_output.put_line ('Addition = ' || (a+b));
    dbms_output.put_line ('Subtraction = ' || (a-b));
    dbms_output.put_line ('Multiplication = ' || (a*b));
    dbms_output.put_line ('Division = ' || (a/b));
    dbms_output.put_line ('Modulus = ' || mod(a,b));
end;
/

```

OUTPUT:

```

Enter value for a: 5
old 1: a number := 5
new 1: a number := 5
Enter value for b: 2
old 2: b number := 2
new 2: b number := 2
Addition = 7
Subtraction = 3
Multiplication = 10
Division = 2.5
Modulus = 1

```

PL/SQL procedure successfully completed.

Q.3. Write a PL/SQL program to input a 5 digit number and display the reverse of that number.

PL/SQL Program :

declare

n Number := '456';

reverse Number := 0;

4 Number;

begin

while (n > 0)

loop

4 := mod (n, 10);

reverse := (reverse * 10) + r;

n := n/10;

end loop;

dbms_output.put_line ('Reverse = ' || reverse);

end;

/

OUTPUT:

Enter value for n: 12345
old 2: n number := &n;
new 2: n number := 12345;
Reverse = 54321

PL/SQL procedure successfully completed.

- Q.4. Write a PL/SQL program to read the radius from the keyboard and insert it into table CIRCLE the radius along with area, the CIRCLE table has two columns defined as radius and area.

PL/SQL Program :

```
declare
    r Number := &r;
    Pi constant number := 3.14;
begin
    insert into circle values (r, pi*r*r);
end;
/
```



OUTPUT:

```
Enter value for r: 5
old  2:  r number := &r;
new  2:  r number := 5;
```

```
Radius and area inserted in circle table
```

```
PL/SQL procedure successfully completed.
```

```
SQL> select * from circle;
```

RADIUS	AREA
5	78.5

DBMS LAB NAME :

Rollno : Section :

- Q.5. Write a PL/SQL program to input the employee number and display its total salary (i.e. sal + comm, and if comm is null assume it to be 0).

PL/SQL Program :

```
declare
    n Number := 1;
begin
    if mod(n, 2) = 0 then
        insert into even values (n);
    else
        insert into odd values (n);
    end if;
end;
```



OUTPUT:

```
Enter value for eno: 7369
old  2:  eno emp.empno%type := &eno;
new  2:  eno emp.empno%type := 7369;
TotalSal = 800
```

PL/SQL procedure successfully completed

Q.6. Write a PL/SQL program to input a number and find whether it is even or odd, if it is even insert it into the table EVEN or insert it into table ODD, both tables have only one column i.e. NUM.

PL/SQL Program:

```

declare
    n Number := 'fn';
begin
    if mod (n,2) = 0 then
        insert into even values (n);
    else
        insert into odd values (n);
    end if;
end;
/

```



OUTPUT:

```

Enter value for n: 5
old    2:  n number := 5;
new    2:  n number := 5;

```

PL/SQL procedure successfully completed.

SQL> select * from even;

no rows selected

SQL> select * from odd;

NUM
5

- Q. Write a PL/SQL program to input a number and find the factorial of that number using
 (i) For loop (ii) Simple loop (iii) While loop

PL/SQL Program using for loop

```

declare
  n Number := 'fn';
  f Number := 1;
begin
  for i in 1...n
    loop
      f := f*i;
    end loop;
  dbms_output.put_line ('Factorial of '||n||' is '||f);
end;
/

```

PL/SQL Program using simple loop.

```

declare
  i Number := 1;
  n Number := 'fn';
  f Number := 1;
begin
  loop
    f := f*i;
    exit when (i >= n);
    i := i+1;
  end loop;
  dbms_output.put_line ('Factorial of '||n||' is '||f);
end;
/

```

PL SQL Program using while loop

declare

i Number := 1;

n Number := '4'n';

f Number := 1;

begin

while (i <= n)

loop

f := f * i;

i := i + 1;

end loop;

dbms_output.put_line ('Factorial of ' || n || ' is ' || f);

end;

/

OUTPUT:

```
Enter value for n: 5
old  2:   n number := 5;
new  2:   n number := 5;
Factorial of 5 is = 120
```

PL/SQL procedure successfully completed.

DBMS LAB

NAME :

Rollno :

Section :

Q.S. Write a PL/SQL program to display the following patterns.

PL/SQL Program :

```

declare
    n Number := 3;
    row Number := 1;
    num Number := 1;
    col Number;

begin
    while (row <= n)
        loop
            col := 1;
            while (col <= row)
                loop
                    dbms_output.put(num);
                    col := col + 1;
                    num := num + 1;
                end loop;
                dbms_output.new_line();
                row := row + 1;
            end loop;
        end;
    /

```

PL/SQL Program :

```

declare
    n Number := 3;
    num Number := 1;
    row Number := 1;
    col Number;

begin
    while (row <= n)
        loop
            while (col <= n)
                loop
                    if (col <= n - row) then
                        dbms_output.put(' ');
                    else
                        dbms_output.put(num);
                        num := num + 1;
                    end if;
                    col := col + 1;
                end loop;
                dbms_output.new_line();
                row := row + 1;
            end loop;
        end;
    /

```

OUTPUT:

1	2	3
4	5	6

1

2

3

4

5

6

PL-SQL Program

declare

nNumber := 2;

nNumber;

nNumber;

col NUMBER;

begin

i := 0;

while (i <= n)

loop

col := 0;

j := 0;

while (j <= n+1)

loop

if (col < n-i) then

dbms_output.put(" ", 2, " ");

else

dbms_output.put(col+i-1 || input(", 2, ' "));

if (j < n) then

col := col + 1;

else

col := col - 1;

end loop;

dbms_output.new_line();

j := j + 1;

end loop;

end;

OUTPUT:

		1		
1	2	3	2	1
1	2	3	2	1

PL/SQL Program :

```

declare
    n Number := 3;
    row Number := 1;
    alpha Number := 65;
    col Number;

begin
    while (row <= n)
        loop
            col := 1;
            while (col <= row)
                loop
                    dbms_output.put (chr(alpha + col - 1));
                    col := col + 1;
                end loop;
                dbms_output.new_line();
                row := row + 1;
            end loop;
        end;
    /

```

OUTPUT:

A		
A	B	
A	B	C

(A+)

RavinderTeacher I/C
Prof. Dinesh Kumar Bhawnani

Steps for explicit cursor

1. Declare a cursor mapped to a SQL select statement that receive data for processing.
2. Open the cursor.
3. Fetch data from the cursor one row at a time into memory variables.
4. Process the data held in the memory variables as required using a loop.
5. Exit from the loop after processing is complete.
6. Closes the cursor.

PL/SQL Programs

Q.1. Write a PL/SQL program to display a message to check whether the record is deleted or not. [Using %found]

PL/SQL Program :

```

declare
    eno Number := eno;
begin
    delete from emp where empno = eno;
    if SQL%found then
        dbms_output.put_line ('Record deleted');
    else
        dbms_output.put_line ('Record not deleted');
    end if;
end;
/

```

OUTPUT:

```

Enter value for eno: 1000
old  2:    delete from emp where empno = &eno;
new  2:    delete from emp where empno = 1000;
Record not deleted

```

```
PL/SQL procedure successfully completed.
```

Q.2. Write a PL/SQL program to display a message to check whether the record is deleted or not. [Using %notfound]

PL/ SQL Program :

declare

eno Number := eno;

begin

delete from emp where empno = eno;

if SQL%not found then

dbms_output.put_line ('Record not deleted');

else

dbms_output.put_line ('Record deleted');

end if;

end %a

/

OUTPUT:

Enter value for eno: 1000
old 2: delete from emp where empno = &eno;
new 2: delete from emp where empno = 1000;
Record not deleted

PL/SQL procedure successfully completed.

Sign of student

- Q.3. Write a PL/SQL program to display a message to give the number of records deleted by the delete statement issued in a PL/SQL block.

PL/SQL Program

```
declare
    eno Number := &eno;
    N Number := 0;
begin
    delete from emp where empno = eno;
    if SQL%found then
        N := SQL%rowcount;
    end if;
    dbms_output.put_line ('Total number of records deleted' || N);
end;
```

OUTPUT:

```
Enter value for eno: 7369
old    4: delete from emp where empno = &eno;
new    4: delete from emp where empno = 7369;
Total number of records deleted 1
```

PL/SQL procedure successfully completed.

Q.4. Write a PL/SQL program to display the empno, ename, job of employees of department number 10.
[Using variables]

PL/SQL Program :

```

declare
    eno emp.empno%type;
    eno empename%type;
    i emp.job%type;
cursor c1 is select empno, ename, job from emp where deptno= 10;
begin
    open c1;
    loop
        fetch c1 into eno, en,j;
        exit when c1%not found;
        dbms_output.put_line('Empno=' || eno);
        dbms_output.put_line('Ename=' || en);
        dbms_output.put_line('Job = ' || j);
    end loop;
    close c1;
end;
/

```

OUTPUT:

empno : 7782
 ename :CLARK
 job :MANAGER
 empno : 7839
 ename :KING
 job :PRESIDENT
 empno : 7934
 ename :MILLER
 job :CLERK

PL/SQL procedure successfully completed.

Q.5. Write a PL/SQL program to display the empno, ename, job of employees of department number 10.
[Using records]

PL/SQL Program :

```

declare
    rec emp%rowtype;
    cursor c1 is select * from emp where deptno=10;
begin
    open c1;
    loop
        fetch c1 into rec;
        exit when c1%notfound;
        dbms_output.put_line ('Empno = ' || rec.empno);
        dbms_output.put_line ('Ename = ' || rec.ename);
        dbms_output.put_line ('Job = ' || rec.job);
    end loop;
    close c1;
end;

```

OUTPUT:

empno : 7782
 ename :CLARK
 job :MANAGER
 empno : 7839
 ename :KING
 job :PRESIDENT
 empno : 7934
 ename :MILLER
 job :CLERK

PL/SQL procedure successfully completed.

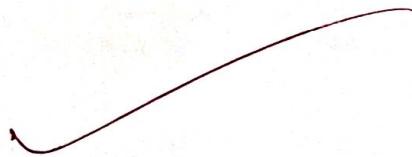
Q.6. Write a PL/SQL program to display the empno, ename, job of employees of department number 10.
 [Using cursor for loop]

PL/SQL Program :

```

declare
  cursor C1 is select * from emp where deptno = 10;
begin
  for rec in C1
  loop
    dbms_output.put_line ('Empno = ' || rec.empno);
    dbms_output.put_line ('Ename = ' || rec.ename);
    dbms_output.put_line ('Job = ' || rec.job);
  end loop,
end;
/

```



OUTPUT:

empno : 7782
 ename :CLARK
 job :MANAGER
 empno : 7839
 ename :KING
 job :PRESIDENT
 empno : 7934
 ename :MILLER
 job :CLERK

PL/SQL procedure successfully completed.

A+
A
10/10

Q.1 Write a PL/SQL program to create a procedure which input a number and find the factorial of a number.

PL/SQL Program :

Create or replace procedure function (n in number out number)

IS

number := 1;

begin

for p in 1...n

loop

m := m * p;

end loop

t := m;

end;

/

OUTPUT:

SQL> variable n number
SQL> execute proc_fact(5,:n);

PL/SQL procedure successfully completed.

SQL> print n

N

120

- Q.2 Write a PL/SQL procedure called MULTI_TABLE that takes two numbers as parameter and displays the multiplication table of the first parameter till the second parameter.

PL/SQL Program :

Create or replace procedure Multi_TABLE (m₁ number, m₂ number)

IS

```

a number := 1;
begin
  for i in 1..m2
    loop
      a := m1 * i;
      dbms_output.put_line (m1' * 'i'; ' = 'a');
    end;
  /

```

OUTPUT:

```

SQL> execute multi_table(2,5)
2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
2 * 4 = 8
2 * 5 = 10

```

PL/SQL procedure successfully completed.

- Q.3 Write a PL/SQL function called POW that takes two numbers as argument and return the value of the first number raised to the power of the second.

PL/SQL Program :

Create or replace ~~finite~~ function pow(a in number, b in number)
return number

IS

power number := 1;

begin

for i in 1...b

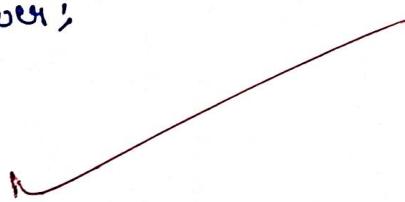
loop

power := power * a;

end loop;

return power;

end;



OUTPUT:
SQL> select pow(2,5) from dual;

POW(2,5)

32

Sign of student

Q.4 Write a PL/SQL program to create a function which input a number and find the factorial of a number.

PL/SQL Program :

Create or replace function fact(n in number return number)

IS

fnumber := 1;

begin

for % in 1...n

loop

f := f * i;

end loop;

return f;

end;

/

OUTPUT:
SQL> select factorial(5) from dual;

FACTORIAL(5)

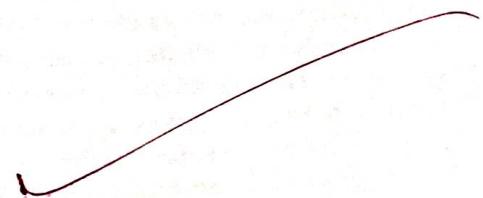
120

Sign of student

Q.5 Write a PL SQL program to create a function that accepts radius of a circle and returns the area of that circle.

PL/SQL Program :

```
create or replace function printarea(a in number) return number
IS
    area number := 1;
begin
    area := 3.14 * r * r;
    return area;
end;
```



OUTPUT:
SQL> select printarea(4) from dual;
PRINTAREA(4)

50.24

AAP
Dinesh Kumar Bhawnani
Teacher I/C

Prof. Dinesh Kumar Bhawnani

Sign of student

DBMS LAB

NAME :

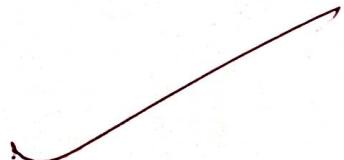
Rollno :

Section :

1. Write a PL/SQL program to create a trigger for the student table which makes the entry in student name column in upper case.

PL/SQL Program :

create or replace trigger tname
before
insert
on student
for each row
begin
 new.sname = upper (!new_sname);
end;
/



OUTPUT :

SQL > insert into student values (1, 'SUMIT', 5, 'cse', 90, 121);

1 row inserted

SQL > select * from student;

ROLLNO SNAME

8 SUMIT

SEM BRANCH

5 cse

MARKS

90

PNO

121

2. Write a PL/SQL program to create a trigger on the student table which shows the old values and new values of student name after updation on student name of student table.

PL/SQL Program :

```
Create or replace trigger tupsname
after
update of sname
on student
for each row
begin
    dbms_output.put_line ('Old value = ' || old.sname);
    dbms_output.put_line ('New value = ' || new.sname);
end;
/
```

OUTPUT :
SQL > update student set sname = 'DINESH' where sname = 'SUMIT';
Old student name is = SUMIT
New student name is = DINESH

Sign of student

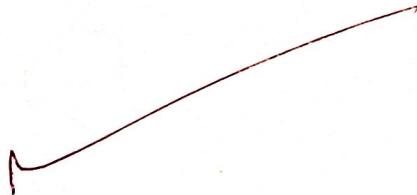
3. Write a PL/SQL program to create a trigger on student table which copies the row which is deleted from student table into student1 table.

PL/SQL Program :

```

Create or replace trigger tdel
before
delete
on student
for each row
begin
    insert into student values (:old.rollno, :old.sname, :old.seno, :old.
Branch, :old.Marks, :old.pno);
end;
/

```



OUTPUT :

SQL > delete from student where rollno = 1;

Old data stored in backup table

SQL > select * from student1;

ROLLNO SNAME

ROLLNO	SNAME	SEM	BRANCH	MARKS	PNO
1 RAM		3	CSE	40	121

Sign of student

4. Write a PL/SQL program to create a trigger that displays a message prior to an insert operation on the student table.

PL/SQL Program :

Create or replace trigger time
before
insert
on student
for each row
begin
dbms_output.put_line('New students are about to be added');
end;
/

OUTPUT :
SQL > insert into student values (7, 'SUMIT', 5, 'CSE', 90, 121);
New students are about to be added
1 row created

Sign of student

5. Write a PL/SQL program to create a trigger for every insert, update and delete operation on the student table, a row is added to the log table recording the date, user and action.

PL/SQL Program :

```

Create or replace trigger trg
before
insert or delete or update
on student
declare out varchar2(20);
begin
  if inserting then
    out := 'INSERT';
  else if deleting then
    out := 'DELETE';
  else if updating then
    out := 'UPDATE';
  else if
    insert into log values (user,out);
end;
  
```

OUTPUT:

```

SQL > insert into student values (2, 'SAURABH', 5, 'CSE', 48, 122);
SQL> select * from logtable;
DT          USR          ACTION
-----  -----
30-JUL-19 DINESH        INSERT
  
```

DBMS LAB

NAME :

Rollno :

Section :

6. Write a PL/SQL program to create a trigger so that no operation can be performed on student table on Sunday.

PL/SQL Program :

```
Create or replace trigger troop
before
insert or delete or update
begin
    if to_char (sysdate , 'd') = 1 then
        raise_application_error (-20200, 'No operation on sunday');
    end if;
end;
```

/

✓

OUTPUT:

```
SQL> delete from student;
delete from student
*
```

```
ERROR at line 1:
ORA-20022: No Operation on Sunday
ORA-06512: at "STUDENT_SUNDAY", line 3
ORA-04088: error during execution of trigger 'STUDENT_SUNDAY'
```

7. Write a PL/SQL program to create a trigger which will verify that no record has the marks value greater than 90 in the student table.

PL/SQL Program :

```
Create or replace trigger to verify
before
insert or update
on student
for each row
begin
    if :new.marks > 90 then
        raise application_error(2000, 'Record is illegal');
    end if;
end;
```

OUTPUT:

```
SQL> update student set marks = 95 where branch = 'CSE';
```

```
update student set marks = 95 where branch = 'CSE'
```

```
*
```

```
ERROR at line 1:
```

```
ORA-20000: Record is illegal
```

```
ORA-06512: at "REC_CHECK", line 3
```

```
ORA-04088: error during execution of trigger 'REC CHECK'
```

8. Write a PL/SQL program which verifies that the updated marks of student is greater than his/ her previous marks.

PL/SQL Program :

```

Create or replace trigger before update update
before
update
on student
for each row
begin
    if :new.marks < :old.marks then
        raise_application_error(20107, 'updated marks is less');
    end if;
end;
/

```

OUTPUT:
SQL> update student set marks = 30 where rollno = 1;
update student set marks = 30 where rollno = 1
*

ERROR at line 1:
ORA-20107: Updated marks is less
ORA-06512: at "TRIGGER7", line 3
ORA-04088: error during execution of trigger 'TRIGGER7'

A.P.P.

Signature
Teacher I/C

Prof. Dinesh Kumar Bhawnani