-- TASK 17

SELECT nickname "Hunts in the field", mice\_ration, bands.name

FROM cats

JOIN bands

ON cats.band\_no = bands.band\_no

WHERE mice\_ration > 50

AND site in ('FIELD', 'WHOLE AREA')

ORDER BY mice\_ration DESC;

-- TASK 18

SELECT cats2.name, cats2.in\_herd\_since "Hunts since"

FROM cats cats1

JOIN cats cats2

ON cats1.name = 'JACEK'

WHERE cats1.in\_herd\_since > cats2.in\_herd\_since

ORDER BY cats2.in\_herd\_since DESC;

-- TASK 19 A

SELECT cats.name, cats.function, cats1.name "Chief 1", cats2.name "Chief 2", cats3.name "Chief 3"

FROM cats

LEFT JOIN cats cats1 ON cats.chief = cats1.nickname

LEFT JOIN cats cats2 ON cats1.chief = cats2.nickname

LEFT JOIN cats cats3 ON cats2.chief = cats3.nickname

WHERE cats.function IN ('CAT', 'NICE');

-- TASK 19 B

SELECT \*

FROM (SELECT CONNECT\_BY\_ROOT name "Name", name chief, CONNECT\_BY\_ROOT function "Function", LEVEL AS "LEV"

FROM cats

CONNECT BY PRIOR chief = nickname

START WITH function IN ('CAT','NICE'))

PIVOT (

MIN(chief)

FOR lev

IN (2 "Chief 1", 3 "Chief 2", 4 "Chief 3")

);

-- TASK 19 C

SELECT CONNECT\_BY\_ROOT name "Name",

CONNECT\_BY\_ROOT function "Function",

LTRIM(SYS\_CONNECT\_BY\_PATH(name, ' | '), ' |') AS "Names of subsequent chiefs"

FROM cats

WHERE chief IS NULL

CONNECT BY PRIOR chief = nickname

START WITH function IN ('CAT','NICE');

-- TASK 20

SELECT

cats.name "Name of female cat",

bands.name "Band name",

enemies.enemy\_name "Enemy name",

enemies.hostility\_degree "Enemy rating",

incident\_date "Incident date"

FROM cats

JOIN bands ON cats.band\_no = bands.band\_no

JOIN incidents ON cats.nickname = incidents.nickname

JOIN enemies ON incidents.enemy\_name = enemies.enemy\_name

WHERE incident\_date > TO\_DATE('2007-01-01')

AND gender = 'W'

ORDER BY cats.name

-- TASK 21

SELECT bands.name, count(DISTINCT cats.nickname)

FROM cats

JOIN bands ON cats.band\_no = bands.band\_no

JOIN incidents ON cats.nickname = incidents.nickname

GROUP BY bands.name;

-- TASK 22

SELECT MIN(function) "Function", cats.nickname "Nickname of cat", count(incidents.nickname) "Number of enemies"

FROM cats

JOIN incidents ON cats.nickname = incidents.nickname

GROUP BY cats.nickname

HAVING count(incidents.nickname) > 1;

-- TASK 23

SELECT name, (mice\_ration + mice\_extra)\*12 "Annual dose", 'above 864' "Dose"

FROM cats

WHERE (mice\_ration + mice\_extra)\*12 > 864 AND mice\_extra IS NOT NULL

UNION

SELECT name, (mice\_ration + mice\_extra)\*12 "Annual dose", '864' "Dose"

FROM cats

WHERE (mice\_ration + mice\_extra)\*12 = 864 AND mice\_extra IS NOT NULL

UNION

SELECT name, (mice\_ration + mice\_extra)\*12 "Annual dose", 'below 864' "Dose"

FROM cats

WHERE (mice\_ration + mice\_extra)\*12 < 864 AND mice\_extra IS NOT NULL

ORDER BY "Annual dose" DESC

-- TASK 24 A

SELECT bands.band\_no, bands.name, site

FROM bands

LEFT JOIN cats ON bands.band\_no = cats.band\_no

WHERE cats.name IS NULL;

-- TASK 24 B

SELECT bands.band\_no, bands.name, site

FROM bands

MINUS

SELECT bands.band\_no, bands.name, site

FROM bands

JOIN cats ON bands.band\_no = cats.band\_no;

-- TASK 25

SELECT name, function, mice\_ration "RATION OF MICE"

FROM cats

WHERE mice\_ration >= ALL(

SELECT 3\*NVL(mice\_ration, 0)

FROM cats

JOIN bands ON cats.band\_no = bands.band\_no

WHERE site in ('WHOLE AREA','ORCHARD')

AND function = 'NICE'

)

-- TASK 26

SELECT function, ROUND(AVG(NVL(mice\_ration, 0) + NVL(mice\_extra,0))) "Average min and max mice"

FROM cats

WHERE function != 'BOSS'

GROUP by function

HAVING

ROUND(AVG(NVL(mice\_ration, 0) + NVL(mice\_extra,0)))

IN(

(

SELECT MIN(ROUND(AVG(NVL(mice\_ration, 0) + NVL(mice\_extra,0))))

FROM cats

WHERE function != 'BOSS'

GROUP by function

),

(

SELECT MAX(ROUND(AVG(NVL(mice\_ration, 0) + NVL(mice\_extra,0))))

FROM cats

WHERE function != 'BOSS'

GROUP by function

)

);

-- TASK 27 A

SELECT nickname, (NVL(mice\_ration, 0) + NVL(mice\_extra, 0)) "EATS"

FROM cats C

WHERE 6 > (

SELECT COUNT(DISTINCT (NVL(mice\_ration, 0) + NVL(mice\_extra, 0)))

FROM cats

WHERE (NVL(C.mice\_ration, 0) + NVL(C.mice\_extra, 0)) < (NVL(mice\_ration, 0) + NVL(mice\_extra, 0))

)

ORDER BY "EATS" DESC;

-- TASK 27 B

SELECT nickname, (NVL(mice\_ration, 0) + NVL(mice\_extra, 0)) "EATS"

FROM cats

WHERE (NVL(mice\_ration, 0) + NVL(mice\_extra, 0)) IN (

SELECT "EATS"

FROM (

SELECT DISTINCT (NVL(mice\_ration, 0) + NVL(mice\_extra, 0)) "EATS"

FROM cats

ORDER BY "EATS" DESC

)

WHERE ROWNUM <= 6

);

-- TASK 27 C

SELECT cats1.nickname, MIN(NVL(cats1.mice\_ration, 0) + NVL(cats1.mice\_extra, 0)) "EATS"

FROM cats cats1

JOIN cats cats2 ON NVL(cats1.mice\_ration,0) + NVL(cats1.mice\_extra,0) <= NVL(cats2.mice\_ration,0) + NVL(cats2.mice\_extra,0)

GROUP BY cats1.nickname

HAVING COUNT(DISTINCT NVL(cats2.mice\_ration,0) +NVL(cats2.mice\_extra,0)) <= 6

ORDER BY "EATS" DESC

-- TASK 28

SELECT TO\_CHAR(EXTRACT(YEAR FROM in\_herd\_since)) "YEAR", COUNT(\*) "NUMBER OF ENTRIES"

FROM cats

GROUP BY EXTRACT(YEAR FROM in\_herd\_since)

HAVING COUNT(\*) IN (

(SELECT MAX(less)

FROM

(SELECT DISTINCT COUNT(\*) less

FROM cats

GROUP BY EXTRACT(YEAR FROM in\_herd\_since)

HAVING COUNT(\*) <

(SELECT AVG(COUNT(EXTRACT(YEAR FROM in\_herd\_since)))

FROM cats

GROUP BY EXTRACT(YEAR FROM in\_herd\_since))

ORDER BY COUNT(\*) DESC

)

),

(SELECT MIN(more)

FROM

(SELECT DISTINCT COUNT(\*) more

FROM cats

GROUP BY EXTRACT(YEAR FROM in\_herd\_since)

HAVING COUNT(\*) >

(SELECT AVG(COUNT(EXTRACT(YEAR FROM in\_herd\_since)))

FROM cats

GROUP BY EXTRACT(YEAR FROM in\_herd\_since))

ORDER BY COUNT(\*))

)

)

UNION ALL

SELECT 'Average', ROUND(AVG(COUNT(nickname)),7)

FROM cats

GROUP BY TO\_CHAR(EXTRACT(YEAR FROM in\_herd\_since))

ORDER BY 2;

-- TASK 29 A

SELECT cats1.name, NVL(cats1.mice\_ration, 0) + NVL(cats1.mice\_extra, 0) "EATS", cats1.band\_no "BAND NO", AVG(NVL(cats2.mice\_ration, 0) + NVL(cats2.mice\_extra, 0)) "AVERAGE"

FROM cats cats1

JOIN cats cats2

ON cats1.band\_no = cats2.band\_no

WHERE cats1.gender = 'M'

GROUP BY cats1.name, cats1.band\_no, NVL(cats1.mice\_ration, 0), NVL(cats1.mice\_extra, 0)

HAVING (NVL(cats1.mice\_ration, 0) + NVL(cats1.mice\_extra, 0)) < AVG(NVL(cats2.mice\_ration, 0) + NVL(cats2.mice\_extra, 0))

ORDER BY cats1.band\_no DESC;

-- TASK 29 B

SELECT cats1.name, NVL(cats1.mice\_ration, 0) + NVL(cats1.mice\_extra, 0) "EATS", cats1.band\_no "BAND NO", AVG "AVERAGE"

FROM (SELECT band\_no bandNo, AVG(mice\_ration + NVL(mice\_extra,0)) "AVG"

FROM cats

GROUP BY band\_no)

JOIN cats cats1

ON cats1.band\_no = bandNo

WHERE cats1.gender = 'M'

AND (NVL(mice\_ration, 0) + NVL(mice\_extra, 0)) < AVG

ORDER BY band\_no DESC;

-- TASK 29 C

SELECT cats1.name, NVL(cats1.mice\_ration, 0) + NVL(cats1.mice\_extra, 0) "EATS", cats1.band\_no "BAND NO",

(

SELECT AVG(NVL(cats2.mice\_ration, 0) + NVL(cats2.mice\_extra, 0))

FROM cats cats2

WHERE cats1.band\_no = cats2.band\_no

) "AVERAGE"

FROM cats cats1

WHERE cats1.gender = 'M'

AND (NVL(cats1.mice\_ration, 0) + NVL(cats1.mice\_extra, 0)) < (

SELECT AVG(NVL(mice\_ration, 0) + NVL(mice\_extra, 0))

FROM cats cats2

WHERE cats1.band\_no = cats2.band\_no

)

ORDER BY band\_no DESC;

-- TASK 30

SELECT c.name, in\_herd\_since "JOIN THE HERD", '<--- SHORTEST TIME IN THE BAND ' || bands.name " "

FROM cats c

JOIN bands

ON c.band\_no = bands.band\_no

WHERE in\_herd\_since = (

SELECT MAX(in\_herd\_since)

FROM cats

WHERE band\_no = c.band\_no

)

UNION ALL

SELECT c.name, in\_herd\_since "JOIN THE HERD", '<--- LONGEST TIME IN THE BAND ' || bands.name " "

FROM cats c

JOIN bands

ON c.band\_no = bands.band\_no

WHERE in\_herd\_since = (

SELECT MIN(in\_herd\_since)

FROM cats

WHERE band\_no = c.band\_no

)

UNION ALL

SELECT c.name, in\_herd\_since "JOIN THE HERD", ' '

FROM cats c

WHERE in\_herd\_since NOT IN (

(

SELECT MIN(in\_herd\_since)

FROM cats

WHERE band\_no = c.band\_no

),

(

SELECT MAX(in\_herd\_since)

FROM cats

WHERE band\_no = c.band\_no)

)

ORDER BY name;

-- TASK 31

DROP VIEW band\_statistics;

CREATE VIEW band\_statistics (BAND\_NAME, AVG\_CONS, MAX\_CONS, MIN\_CONS, CAT, CAT\_WITH\_EXTRA) AS

SELECT

b.name,

AVG(c.mice\_ration),

MAX(c.mice\_ration),

MIN(c.mice\_ration),

COUNT(\*),

COUNT(c.mice\_extra)

FROM bands b

JOIN cats c ON b.band\_no = c.band\_no

GROUP BY b.name;

SELECT \*

FROM band\_statistics

ORDER BY 2 DESC;

SELECT

c.nickname,

c.name,

c.function,

c.mice\_ration "EATS",

'FROM ' || bs.min\_cons ||

' TO ' || bs.max\_cons "CONSUMPTION LIMITS",

c.in\_herd\_since "HUNT FROM"

FROM

band\_statistics bs

JOIN

bands b ON bs.band\_name = b.name

JOIN

cats c ON b.band\_no = c.band\_no

WHERE

c.nickname = 'CAKE';

-- TASK 32

SELECT nickname "Nickname", gender "Gender", mice\_ration "Mice before pay increase", NVL(mice\_extra,0) "Extra before pay increase"

FROM Cats

LEFT JOIN bands b on CATS.BAND\_NO = b.BAND\_NO

WHERE nickname IN (SELECT nickname --po nickname?

FROM (SELECT nickname

FROM CATS

LEFT JOIN BANDS b USING (BAND\_NO)

WHERE b.name = 'BLACK KNIGHTS'

ORDER BY IN\_HERD\_SINCE)

WHERE ROWNUM <=3

UNION ALL

SELECT nickname --po nickname?

FROM (SELECT nickname

FROM CATS

LEFT JOIN BANDS b USING (BAND\_NO)

WHERE b.name = 'PINTO HUNTERS'

ORDER BY IN\_HERD\_SINCE)

WHERE ROWNUM <=3);

--update

UPDATE Cats

SET mice\_ration = CASE gender

WHEN 'W' THEN mice\_ration + (SELECT MIN(mice\_ration)

FROM CATS) \* 0.10

WHEN 'M' THEN mice\_ration + 10

END,

mice\_extra = NVL(mice\_extra,0) + (SELECT AVG(NVL(mice\_extra, 0))

FROM CATS c

WHERE c.BAND\_NO = Cats.band\_no) \* 0.15

WHERE nickname IN (SELECT nickname

FROM (SELECT nickname

FROM CATS

LEFT JOIN BANDS b USING (BAND\_NO)

WHERE b.name = 'BLACK KNIGHTS'

ORDER BY IN\_HERD\_SINCE)

WHERE ROWNUM <=3

UNION ALL

SELECT nickname

FROM (SELECT nickname

FROM CATS

LEFT JOIN BANDS b USING (BAND\_NO)

WHERE b.name = 'PINTO HUNTERS'

ORDER BY IN\_HERD\_SINCE)

WHERE ROWNUM <=3);

SELECT nickname "Nickname", gender "Gender", mice\_ration "Mice after pay increase", NVL(mice\_extra,0) "Extra after pay increase"

FROM Cats

LEFT JOIN bands b on CATS.BAND\_NO = b.BAND\_NO

WHERE nickname IN (SELECT nickname --po nickname?

FROM (SELECT nickname

FROM CATS

LEFT JOIN BANDS b USING (BAND\_NO)

WHERE b.name = 'BLACK KNIGHTS'

ORDER BY IN\_HERD\_SINCE)

WHERE ROWNUM <=3

UNION ALL

SELECT nickname --po nickname?

FROM (SELECT nickname

FROM CATS

LEFT JOIN BANDS b USING (BAND\_NO)

WHERE b.name = 'PINTO HUNTERS'

ORDER BY IN\_HERD\_SINCE)

WHERE ROWNUM <=3);

ROLLBACK;