

練習問題

目次

1. ユーザーと権限	1
2. テーブル操作	2
3. レコードの登録	5
4. レコードの検索	7
5. レコードの更新	10
6. レコードの削除	11
7. トランザクションとロック	12
8. 演算子	14
9. 集約関数	17
10. グループ化	18
11. 並び替え	20
12. 結合	21
13. サブクエリ	26
14. シーケンス	28
15. 正規化	29
16. JDBC	32

1. ユーザーと権限

- (1) ユーザーを作成しなさい。ただし、ユーザー名は education、パスワードは systemsss とする。※作成する前に、system ユーザーで「ALTER SESSION SET CONTAINER = xepdb1;」のコマンドを実行してセッションを PDB に切り替えること。

```
CREATE USER education IDENTIFIED BY systemsss;
```

- (2) education ユーザーに対し、SELECT ANY DECTIONARY 権限以外の全ての権限を付与しなさい。

```
GRANT ALL PRIVILEGES TO education;
```

2. テーブル操作

(1) education スキーマ内にテーブルを作成しなさい。テーブル名は dept とし、定義の詳細については下記を参考にする事。

dept テーブル

No	論理名称	物理名称	データ型	桁数	制約
1	部署 NO	deptno	VARCHAR2	2	PRIMARY KEY
2	部署名	dname	VARCHAR2	50	NOT NULL
3	場所	loc	VARCHAR2	100	

```
CREATE TABLE dept(  
    deptno VARCHAR2(2) PRIMARY KEY,  
    dname VARCHAR2(50) NOT NULL,  
    loc VARCHAR2(100)  
);
```

(2) education スキーマ内にテーブルを作成しなさい。テーブル名は emp とし、定義の詳細については下記を参考にすること。なお、deptno カラムは dept テーブルの deptno カラムと関連付け、外部参照制約については表制約で設定し、制約名は fk_deptno とすること。

emp テーブル

No	論理名称	物理名称	データ型	桁数	制約
1	社員 NO	empno	VARCHAR2	3	PRIMARY KEY
2	社員名	ename	VARCHAR2	20	NOT NULL
3	上司	superior	VARCHAR2	20	
4	入社日	hiredate	DATE		NOT NULL
5	給与	sal	NUMBER		NOT NULL
6	職種	job	VARCHAR2	30	
7	部署 NO	deptno	VARCHAR2	2	FOREIGN KEY

```
CREATE TABLE emp (  
    empno VARCHAR2(3) PRIMARY KEY,  
    ename VARCHAR2(20) NOT NULL,  
    superior VARCHAR2(20),  
    hiredate DATE NOT NULL,  
    sal NUMBER NOT NULL,  
    job VARCHAR2(30),  
    deptno VARCHAR2(2),  
    CONSTRAINT fk_deptno FOREIGN KEY(deptno)  
        REFERENCES dept(deptno)  
);
```

(3) 上記(2)で作成した emp テーブルを作成する SQL 文に、適切なコメントを付与しなさい。
少なくとも各カラムには論理名称を 1 行コメントにて付加し、
表制約に対しては複数行コメントを利用し、詳細な説明を付加すること。

```
CREATE TABLE emp (  
    empno VARCHAR2(3) PRIMARY KEY,      -- 社員 NO  
    ename VARCHAR2(20) NOT NULL,        -- 社員名  
    superior VARCHAR2(20),              -- 上司  
    hiredate DATE NOT NULL,              -- 入社日  
    sal NUMBER NOT NULL, -- 給与  
    job VARCHAR2(30) ,    -- 職種  
    deptno VARCHAR2(2),    -- 部署 NO  
    /*  
    * 部署テーブルの deptno（部署 NO） カラムを  
    * 外部参照制約する制約を付加する。  
    */  
    CONSTRAINT fk_deptno FOREIGN KEY(deptno)  
        REFERENCES dept(deptno)  
);
```

3. レコードの登録

(1) emp テーブル及び dept テーブルにデータを登録しなさい。詳細については下記を参考にすること。

emp テーブル

empno	ename	superior	hiredate	sal	job	deptno
001	鈴木 仁	NULL	2000/01/01	600000	PL	01
002	佐藤 幸助	鈴木 仁	2000/03/22	350000	ブリッジ SE	01
003	高橋 ひとみ	NULL	2000/01/01	500000	PL	02
004	大前 貢	鈴木 仁	2000/04/01	320000	SE	01
005	真田 博之	鈴木 仁	2001/04/01	300000	SE	01
006	大城 勉	鈴木 仁	2001/04/01	300000	SE	01
007	神城 陣	高橋 ひとみ	2001/04/01	280000	PG	02
008	結城 瞳	高橋 ひとみ	2002/04/01	280000	PG	02
009	博多 弁	高橋 ひとみ	2003/04/01	260000	PG	02
010	小山 浩	高橋 ひとみ	2006/04/01	180000	デザイナー	02

dept テーブル

deptno	dname	loc
01	システム企画本部	東京都江東区豊洲 3-3-3
02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1
03	営業部	大阪府大阪市中央区城見 1-4-24
04	総務部	NULL

-- dept テーブル

```
INSERT INTO dept VALUES ('01','システム企画本部','東京都江東区豊洲 3-3-3');
INSERT INTO dept VALUES ('02','ソリューション事業部','神奈川県川崎市中原区
上小田中 4-1-1');
INSERT INTO dept VALUES ('03','営業部','大阪府大阪市中央区城見 1-4-24');
INSERT INTO dept VALUES ('04','総務部',NULL);
```

COMMIT;

次ページに続く

```
-- emp
INSERT INTO emp VALUES ('001', '鈴木 仁', NULL, '2000/01/01', 600000, 'PL',
'01');
INSERT INTO emp VALUES ('002', '佐藤 幸助', '鈴木 仁', '2000/03/22', 350000, '
ブリッジ SE', '01');
INSERT INTO emp VALUES ('003', '高橋 ひとみ', NULL, '2000/01/01', 500000,
'PL', '02');
INSERT INTO emp VALUES ('004', '大前 貢', '鈴木 仁', '2000/04/01', 320000,
'SE', '01');
INSERT INTO emp VALUES ('005', '真田 博之', '鈴木 仁', '2001/04/01',
300000, 'SE', '01');
INSERT INTO emp VALUES ('006', '大城 勉', '鈴木 仁', '2001/04/01', 300000,
'SE', '01');
INSERT INTO emp VALUES ('007', '神城 陣', '高橋 ひとみ', '2001/04/01',
280000, 'PG', '02');
INSERT INTO emp VALUES ('008', '結城 瞳', '高橋 ひとみ', '2002/04/01',
280000, 'PG', '02');
INSERT INTO emp VALUES ('009', '博多 弁', '高橋 ひとみ', '2003/04/01',
260000, 'PG', '02');
INSERT INTO emp VALUES ('010', '小山 浩', '高橋 ひとみ', '2006/04/01',
180000, 'デザイナー', '02');

COMMIT;
```

4. レコードの検索

(1) dept テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

DEPTNO	DNAME	LOC
01	システム企画本部	東京都江東区豊洲 3-3-3
02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1
03	営業部	大阪府大阪市中央区城見 1-4-24
04	総務部	NULL

```
SELECT * FROM dept;
```

(2) emp テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

ENAME	SUPERIOR	HIREDATE
鈴木 仁	NULL	00-01-01
佐藤 幸助	鈴木 仁	00-03-22
高橋 ひとみ	NULL	00-01-01
大前 貢	鈴木 仁	00-04-01
真田 博之	鈴木 仁	01-04-01
大城 勉	鈴木 仁	01-04-01
神城 陣	高橋 ひとみ	01-04-01
結城 瞳	高橋 ひとみ	02-04-01
博多 弁	高橋 ひとみ	03-04-01
小山 浩	高橋 ひとみ	06-04-01

```
SELECT ename, superior, hiredate FROM emp;
```


(3) emp テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

社員名	上司	給与

鈴木 仁	NULL	600000
佐藤 幸助	鈴木 仁	350000
高橋 ひとみ	NULL	500000
大前 貢	鈴木 仁	320000
真田 博之	鈴木 仁	300000
大城 勉	鈴木 仁	300000
神城 陣	高橋 ひとみ	280000
結城 瞳	高橋 ひとみ	280000
博多 弁	高橋 ひとみ	260000
小山 浩	高橋 ひとみ	180000

```
SELECT ename AS "社員名", superior AS "上司", sal AS "給与" FROM emp;
```

(4) emp テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

JOB

ブリッジ SE
デザイナー
SE
PG
PL

```
SELECT DISTINCT job FROM emp;
```

(5) dept テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

DEPTNO	DNAME	LOC

02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1

```
SELECT * FROM dept WHERE deptno = '02';
```

(6) emp テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

EMP NO	ENAME	HIREDATE	JOB

007	神城 陣	01-04-01	PG
008	結城 瞳	02-04-01	PG
009	博多 弁	03-04-01	PG

```
SELECT empno, ename, hiredate, job FROM emp WHERE job = 'PG';
```

5. レコードの更新

(1) dept テーブルのデータを更新しなさい。詳細については下記を参考にすること。

dept テーブル

deptno	dname	loc
01	システム企画本部	東京都江東区豊洲 3-3-3
02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1
03	営業部	大阪府大阪市中央区城見 1-4-24
04	人事部	NULL

```
UPDATE dept SET dname = '人事部' WHERE deptno = '04';

COMMIT;
```

(2) emp テーブルのデータを更新しなさい。詳細については下記を参考にすること。

emp テーブル

empno	ename	superior	hiredate	sal	job	deptno
001	鈴木 仁	NULL	2000/01/01	600000	PL	01
002	佐藤 幸助	鈴木 仁	2000/03/22	350000	ブリッジ SE	01
003	高橋 ひとみ	NULL	2000/01/01	500000	PL	02
004	大前 貢	鈴木 仁	2000/04/01	320000	SE	01
005	真田 博之	鈴木 仁	2001/04/01	300000	SE	01
006	大城 勉	鈴木 仁	2001/04/01	300000	SE	01
007	神城 陣	高橋 ひとみ	2001/04/01	280000	PG	02
008	結城 瞳	鈴木 仁	2002/04/01	290000	SE	01
009	博多 弁	高橋 ひとみ	2003/04/01	260000	PG	02
010	小山 浩	高橋 ひとみ	2006/04/01	180000	デザイナー	02

```
UPDATE emp SET superior = '鈴木 仁', job = 'SE', sal = 290000,
deptno = '01' WHERE empno = '008';

COMMIT;
```

6. レコードの削除

(1) emp テーブルから empno が '006' のデータを削除して、下記のような結果になるように
しなさい。

emp テーブル

empno	ename	superior	hiredate	sal	job	deptno
001	鈴木 仁	NULL	2000/01/01	600000	PL	01
002	佐藤 幸助	鈴木 仁	2000/03/22	350000	ブリッジ SE	01
003	高橋 ひとみ	NULL	2000/01/01	500000	PL	02
004	大前 貢	鈴木 仁	2000/04/01	320000	SE	01
005	真田 博之	鈴木 仁	2001/04/01	300000	SE	01
007	神城 陣	高橋 ひとみ	2001/04/01	280000	PG	02
008	結城 瞳	鈴木 仁	2002/04/01	290000	SE	01
009	博多 弁	高橋 ひとみ	2003/04/01	260000	PG	02
010	小山 浩	高橋 ひとみ	2006/04/01	180000	デザイナー	02

```
DELETE FROM emp WHERE empno = '006';
```

```
COMMIT;
```

7. トランザクションとロック

(1) dept テーブルにデータ（下表の赤字のレコード）を登録し、その後データの登録を取り消しなさい。詳細については下記を参考にすること。

dept テーブル

deptno	dname	loc
01	システム企画本部	東京都江東区豊洲 3-3-3
02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1
03	営業部	大阪府大阪市中央区域見 1-4-24
04	人事部	NULL
20	経理部	Redmond, WA 98052-7329 USA
21	水産食品開発部	NULL
22	食品総合研究所	NULL

```
INSERT INTO dept VALUES ('20','経理部','Redmond, WA 98052-7329 USA');
INSERT INTO dept VALUES ('21','水産食品開発部',NULL);
INSERT INTO dept VALUES ('22','食品総合研究所',NULL);

ROLLBACK;
```

(2) dept テーブルにデータを登録し、その後 deptno カラムの値が 21、22 のデータの登録を取り消しなさい。なお、deptno カラムの値が 20 のデータの登録は確定させなさい。詳細については下記を参考にし、データの登録の取り消し、確定には SAVEPOINT を用いること。

dept テーブル

deptno	dname	loc
01	システム企画本部	東京都江東区豊洲 3-3-3
02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1
03	営業部	大阪府大阪市中央区城見 1-4-24
04	人事部	NULL
20	経理部	Redmond, WA 98052-7329 USA
21	水産食品開発部	NULL
22	食品総合研究所	NULL

```
INSERT INTO dept VALUES ('20','経理部','Redmond, WA 98052-7329 USA');
```

```
SAVEPOINT sp1;
```

```
INSERT INTO dept VALUES ('21','水産食品開発部',NULL);
```

```
INSERT INTO dept VALUES ('22','食品総合研究所',NULL);
```

```
ROLLBACK TO sp1;
```

```
COMMIT;
```

8. 演算子

(1) empテーブルからデータを検索しなさい。上司がNULLでないデータを検索するものとし、下記の実行結果を参考にする。

実行結果

ENAME	SUPERIOR

佐藤 幸助	鈴木 仁
大前 貢	鈴木 仁
真田 博之	鈴木 仁
神城 陣	高橋 ひとみ
結城 瞳	鈴木 仁
博多 弁	高橋 ひとみ
小山 浩	高橋 ひとみ

```
SELECT ename, superior FROM emp WHERE superior IS NOT NULL;
```

(2) emp テーブルからデータを検索しなさい。給与が 300000 以下のデータを検索するものとし、下記の実行結果を参考にする。

実行結果

ENAME	SAL	JOB

真田 博之	300000	SE
神城 陣	280000	PG
結城 瞳	290000	SE
博多 弁	260000	PG
小山 浩	180000	デザイナー

```
SELECT ename, sal, job FROM emp WHERE sal <= 300000;
```

(3) emp テーブルからデータを検索しなさい。職種が SE かつ部署 NO が 01 のデータを検索するものとし、下記の実行結果を参考にすること。

実行結果

ENAME	JOB	DEPTNO

大前 貢	SE	01
真田 博之	SE	01
結城 瞳	SE	01

```
SELECT ename, job, deptno FROM emp WHERE job = 'SE' AND deptno = '01';
```

(4) emp テーブルからデータを検索しなさい。職種が PL または部署 NO が 01 のデータを検索するものとし、下記の実行結果を参考にすること。

実行結果

ENAME	JOB	DEPTNO

鈴木 仁	PL	01
佐藤 幸助	ブリッジ SE	01
高橋 ひとみ	PL	02
大前 貢	SE	01
真田 博之	SE	01
結城 瞳	SE	01

```
SELECT ename, job, deptno FROM emp WHERE job = 'PL' OR deptno = '01';
```


(5) emp テーブルからデータを検索しなさい。社員名に“城”の文字が含まれているデータを検索するものとし、下記の実行結果を参考にする。

実行結果

EMP NO	ENAME

007	神城 陣
008	結城 瞳

```
SELECT empno, ename FROM emp WHERE ename LIKE '%城%';
```

(6) emp テーブルからデータを検索しなさい。入社日が 2000 年 1 月 1 日から 2001 年 12 月 31 日までのデータを検索するものとし、下記の実行結果を参考にする。

実行結果

ENAME	HIREDATE

鈴木 仁	00-01-01
佐藤 幸助	00-03-22
高橋 ひとみ	00-01-01
大前 貢	00-04-01
真田 博之	01-04-01
神城 陣	01-04-01

```
SELECT ename, hiredate FROM emp WHERE hiredate BETWEEN '2000/01/01'  
AND '2001/12/31';
```

9. 集約関数

(1) emp テーブルからデータを検索しなさい。emp テーブルの行数を検索するものとし、下記の実行結果を参考にすること。

実行結果

行数

9

```
SELECT COUNT(*) AS "行数" FROM emp;
```

(2) emp テーブルからデータを検索しなさい。給与の最大値、最小値、合計値を検索するものとし、下記の実行結果を参考にすること。

実行結果

給与(最大値)	給与(最小値)	給与(合計値)
-----	-----	-----
600000	180000	3080000

```
SELECT MAX(sal) AS "給与(最大値)", MIN(sal) AS "給与(最小値)", SUM(sal) AS "給与(合計値)" FROM emp;
```

(3)集約関数を利用して、入社日のデータのなかで最も新しい日付を検索しなさい。

実行結果

入社日

06-04-01

```
SELECT MAX(hiredate) AS "入社日" FROM emp;
```

10. グループ化

(1) emp テーブルからデータを検索しなさい。職種に対する行数を検索するものとし、下記の
実行結果を参考にすること。

実行結果

JOB	行数

ブリッジ SE	1
デザイナー	1
SE	3
PG	2
PL	2

```
SELECT job, COUNT(*) AS "行数" FROM emp GROUP BY job;
```

(2) emp テーブルからデータを検索しなさい。上司に対する行数を検索するものとし、下記の

実行結果を参考にすること。

実行結果

SUPERIOR	行数

鈴木 仁	4
高橋 ひとみ	3

```
SELECT superior, COUNT(*) AS "行数" FROM emp WHERE superior  
IS NOT NULL GROUP BY superior;
```

(3) emp テーブルからデータを検索しなさい。職種毎の平均給与を表示するものとし、下記の
実行結果を参考にすること。

実行結果

JOB	平均給与
-----	-----
ブリッジ SE	350000
デザイナー	180000
SE	303333.33
PG	270000
PL	550000

```
SELECT job , avg(sal) AS “平均給与” FROM EMP GROUP BY job;
```

11. 並び替え

(1) emp テーブルからデータを検索しなさい。下記の実行結果を参考にする事。

実行結果

EMPNO	ENAME

010	小山 浩
009	博多 弁
008	結城 瞳
007	神城 陣
005	真田 博之
004	大前 貢
003	高橋 ひとみ
002	佐藤 幸助
001	鈴木 仁

```
SELECT empno, ename FROM emp ORDER BY empno DESC;
```

(2) emp テーブルからデータを検索し、DEPTNO 昇順、SAL 降順で表示しなさい。下記の実行結果を参考にする事。

実行結果

DEPTNO	ENAME	SAL

01	鈴木 仁	600000
01	佐藤 幸助	350000
01	大前 貢	320000
01	真田 博之	300000
01	結城 瞳	290000
02	高橋 ひとみ	500000
02	神城 陣	280000
02	博多 弁	260000
02	小山 浩	180000

```
SELECT deptno,ename,sal FROM emp ORDER BY deptno ASC, sal DESC;
```

12. 結合

(1) dept テーブル及び emp テーブルからデータを検索しなさい。下記の実行結果を参考にする
こと。

実行結果

EMP NO	ENAME	DNAME

001	鈴木 仁	システム企画本部
002	佐藤 幸助	システム企画本部
003	高橋 ひとみ	ソリューション事業部
004	大前 貢	システム企画本部
005	真田 博之	システム企画本部
007	神城 陣	ソリューション事業部
008	結城 瞳	システム企画本部
009	博多 弁	ソリューション事業部
010	小山 浩	ソリューション事業部

```
SELECT
    e.empno,
    e.ename,
    d.dname
FROM
    emp e INNER JOIN dept d
ON    e.deptno = d.deptno;
```

(2) dept テーブル及び emp テーブルからデータを検索しなさい。下記の実行結果を参考にする
こと。

実行結果

EMP NO	ENAME	DNAME

001	鈴木 仁	システム企画本部
002	佐藤 幸助	システム企画本部
003	高橋 ひとみ	ソリューション事業部
004	大前 貢	システム企画本部
005	真田 博之	システム企画本部
007	神城 陣	ソリューション事業部
008	結城 瞳	システム企画本部
009	博多 弁	ソリューション事業部
010	小山 浩	ソリューション事業部
NULL	NULL	経理部
NULL	NULL	営業部
NULL	NULL	人事部

```
SELECT
    e.empno,
    e.ename,
    d.dname
FROM
    emp e RIGHT OUTER JOIN dept d
ON    e.deptno = d.deptno;
```

(3) education スキーマ内にテーブルを作成しなさい。テーブル名は plan とし、定義の詳細については下記を参考にすること。なお、empno カラムは emp テーブルの empno カラムと関連付けなさい。

plan テーブル

No	論理名称	物理名称	データ型	桁数	制約
1	予定 NO	planno	VARCHAR2	4	PRIMARY KEY
2	社員 NO	empno	VARCHAR2	3	FOREIGN KEY
3	予定日	scheduled_date	DATE		NOT NULL
4	予定	plan	VARCHAR2	200	NOT NULL

```
CREATE TABLE plan(
    planno VARCHAR2(4) PRIMARY KEY,
    empno VARCHAR2(3),
    scheduled_date DATE NOT NULL,
    plan VARCHAR2(200) NOT NULL,
    FOREIGN KEY(empno) REFERENCES emp(empno)
);
```

(4) plan テーブルにデータを登録しなさい。詳細については下記を参考にすること。

plan テーブル

planno	empno	scheduled_date	plan
0001	001	2007/02/03	14 時から A 社様との面談
0002	003	2007/02/03	大阪出張
0003	003	2007/02/05	大阪出張
0004	001	2007/02/04	9 時 30 分から進捗報告会
0005	001	2007/02/06	18 時から経営会議

```
-- PLAN
INSERT INTO plan VALUES ('0001','001','2007/02/03','14 時から A 社様との面談');
INSERT INTO plan VALUES ('0002','003','2007/02/03','大阪出張');
INSERT INTO plan VALUES ('0003','003','2007/02/05','大阪出張');
INSERT INTO plan VALUES ('0004','001','2007/02/04','9 時 30 分から進捗報告会');
INSERT INTO plan VALUES ('0005','001','2007/02/06','18 時から経営会議');

COMMIT;
```


(5) dept テーブル、emp テーブル、plan テーブルからデータを検索しなさい。下記の実行結果を参考にする。

実行結果

PLANNO	SCHEDULED_DATE	PLAN	EMP NO	ENAME	DNAME
0001	07-02-03	14 時から A 社様との面談	001	鈴木 仁	システム企画本部
0002	07-02-03	大阪出張	003	高橋 ひとみ	ソリューション事業部
0003	07-02-05	大阪出張	003	高橋 ひとみ	ソリューション事業部
0004	07-02-04	9 時 30 分から進捗報告会	001	鈴木 仁	システム企画本部
0005	07-02-06	18 時から経営会議	001	鈴木 仁	システム企画本部

```
SELECT
  p.planno,
  p.scheduled_date,
  p.plan,
  e.empno,
  e.ename,
  d.dname
FROM
  plan p INNER JOIN emp e
    ON p.empno = e.empno
  INNER JOIN dept d
    ON e.deptno = d.deptno
ORDER BY p.planno;
```

(6) dept テーブル、emp テーブル、plan テーブルからデータを検索しなさい。plan テーブルもレコード件数を検索するものとし、下記の実行結果を参考にする。

実行結果

EMP NO	ENAME	DNAME	件数

001	鈴木 仁	システム企画本部	3
003	高橋 ひとみ	ソリューション事業部	2

```
SELECT
    p.empno,
    e.ename,
    d.dname,
    COUNT(*) AS "件数"
FROM
    emp e INNER JOIN plan p
    ON e.empno = p.empno
    INNER JOIN dept d
    ON e.deptno = d.deptno
GROUP BY p.empno, e.ename, d.dname
ORDER BY p.empno;
```

13. サブクエリ

(1) emp テーブルからデータを検索しなさい。給与が平均以上のデータを検索するものとし、下記の実行結果を参考にする。

実行結果

EMP NO	ENAME	SAL

001	鈴木 仁	600000
002	佐藤 幸助	350000
003	高橋 ひとみ	500000

```
SELECT
    empno, ename, sal
FROM
    emp
WHERE
    sal >= (
        SELECT AVG(sal) FROM emp
    );
```

(2) emp テーブルからデータを検索しなさい。同じ職種が 2 人以上存在するデータを検索するものとし、下記の実行結果を参考にすること。

実行結果

EMP NO	ENAME	JOB

001	鈴木 仁	PL
003	高橋 ひとみ	PL
004	大前 貢	SE
005	真田 博之	SE
007	神城 陣	PG
008	結城 瞳	SE
009	博多 弁	PG

```
--解答例 1
SELECT
    empno, ename, job
FROM
    emp
WHERE
    job IN (SELECT job
            FROM emp
            GROUP BY job
            HAVING COUNT(*) >= 2
    );
```

```
--解答例 2
SELECT
    e1.empno, e1.ename, e1.job
FROM
    emp e1 INNER JOIN (
        SELECT job
        FROM emp
        GROUP BY job
        HAVING COUNT(*) >= 2
    ) e2
    ON e1.job = e2.job
ORDER BY e1.empno;
```

14. シーケンス

(1) education スキーマ内にシーケンスを作成しなさい。シーケンス名は dept_seq とし、初期値は 10、増減値は 1 としなさい。

```
CREATE SEQUENCE dept_seq  
START WITH 10  
INCREMENT BY 1;
```

(2) dept テーブルにデータを登録しなさい。詳細については下記を参考にする。なお、deptno カラムに格納する値については、dept_seq シーケンスを使用すること。

dept テーブル

deptno	dname	loc
01	システム企画本部	東京都江東区豊洲 3-3-3
02	ソリューション事業部	神奈川県川崎市中原区上小田中 4-1-1
03	営業部	大阪府大阪市中央区城見 1-4-24
04	人事部	NULL
20	経理部	Redmond, WA 98052-7329 USA
10	経営統括部	Redmond, WA 98052-7329 USA
11	ネットワーク事業部	NULL

```
INSERT INTO dept VALUES (dept_seq.NEXTVAL, '経営統括部', 'Redmond, WA  
98052-7329 USA');  
INSERT INTO dept VALUES (dept_seq. NEXTVAL, 'ネットワーク事業部', NULL);  
  
COMMIT;
```

15. 正規化

キャラ ID	キャラクター	スキル ID1	スキル 1	カテゴリ 1	ス キ ル ID2	スキル 2	カテゴリ 2
1	勇者	1	弓を放つ	攻撃	2	盾を使う	防御
2	魔法使い	3	空を飛ぶ	防御	4	傷を癒す	回復
3	狩人	1	弓を放つ	攻撃	5	逃げる	逃亡
4	妖精	6	姿を消す	逃亡	3	空を飛ぶ	防御
5	犬	7	体当たり	攻撃	5	逃げる	逃亡

(1) 上のテーブルを講義資料の説明に沿った形で第 1 正規化して下さい。

キャラ ID	キャラクター	スキル ID	スキル	カテゴリ
1	勇者	1	弓を放つ	攻撃
1	勇者	2	盾を使う	防御
2	魔法使い	3	空を飛ぶ	防御
2	魔法使い	4	傷を癒す	回復
3	狩人	1	弓を放つ	攻撃
3	狩人	5	逃げる	逃亡
4	妖精	6	姿を消す	逃亡
4	妖精	3	空を飛ぶ	防御
5	犬	7	体当たり	攻撃
5	犬	5	逃げる	逃亡

(2) (1) のテーブルを講義資料の説明に沿った形で第 2 正規化しなさい。
テーブル間のレコードの関係性が保持できるようにすること。

キャラクターテーブル

キャラ ID	キャラクター
1	勇者
2	魔法使い
3	狩人
4	妖精
5	犬

スキル・カテゴリテーブル

スキル ID	スキル	カテゴリ
1	弓を放つ	攻撃
2	盾を使う	防御
3	空を飛ぶ	防御
4	傷を癒す	回復
5	逃げる	逃亡
6	姿を消す	逃亡
7	体当たり	攻撃

保持スキル管理テーブル

保持スキル管理 ID	キャラ ID	スキル ID
1	1	1
2	1	2
3	2	3
4	2	4
5	3	1
6	3	5
7	4	6
8	4	3
9	5	7
10	5	5

(3) (2) のテーブルを講義資料の説明に沿った形で第 3 正規化しなさい。
必要に応じて列を追加しても良いとする。

キャラクターテーブル

キャラ ID	キャラクター
1	勇者
2	魔法使い
3	狩人
4	妖精
5	犬

スキルテーブル

スキル ID	スキル	カテゴリ ID
1	弓を放つ	1
2	盾を使う	2
3	空を飛ぶ	2
4	傷を癒す	3
5	逃げる	4
6	姿を消す	4
7	体当たり	1

カテゴリテーブル

カテゴリ ID	カテゴリ
1	攻撃
2	防御
3	回復
4	逃亡

保持スキル管理テーブル

保持スキル管理 ID	キャラ ID	スキル ID
1	1	1
2	1	2
3	2	3
4	2	4
5	3	1
6	3	5
7	4	6
8	4	3
9	5	7
10	5	5

16. JDBC

※JDBC の練習問題の解答は Java プロジェクトとして配布致します。
LMS から以下のファイルをダウンロードしてください。

oracle_execises_answer.zip