Phase 2 Di Zhao 47509882

Part 1: Create Tables:

```sql
create table SOFTWARE_PRODUCT
(
Pname varchar(20) not null,
Pversion varchar(20) not null,
Pstatus enum('ready', 'usable', 'not-ready') default "not-ready",
primary key(Pname, Pversion)
);

create table COMPONENT
(
Cname varchar(50) not null,
Cversion varchar(3) not null,
Language enum('C','C++','C#','Java','PHP'),
Size integer not null,
Cstatus enum('ready', 'usable', 'not-ready') default "not-ready",
primary key(Cname, Cversion)
);

create table EMPLOYEE
(
ID integer not null,
Ename varchar(60) not null,
Hiredate date not null,
Mgr integer,
Seniority varchar(10) default null,
primary key(ID)
);

create table INSPECTION
(
Cname varchar(50) not null,
Cversion varchar(3) not null,
Date date not null,
Score integer,
Description varchar(4000) not null,
key(date),
foreign key(Cname, Cversion) references COMPONENT(Cname, Cversion) on delete cascade on update cascade
);
```

```sql
create table HAVE
(
Pname varchar(10) not null,
Pversion varchar(10) not null,
Cname varchar(50) not null,
Cversion varchar(3) not null,
foreign key(Pname, Pversion) references SOFTWARE_PRODUCT(Pname, Pversion) on delete cascade on update cascade,
foreign key(Cname, Cversion) references COMPONENT(Cname, Cversion) on delete cascade on update cascade
);

create table OWN
(
Cname varchar(50) not null,
Cversion varchar(3) not null,
ID integer not null,
foreign key(Cname, Cversion) references COMPONENT(Cname, Cversion) on delete cascade on update cascade,
foreign key(ID) references EMPLOYEE(ID) on delete cascade on update cascade
);

create table INSPECTED
(
Cname varchar(50) not null,
Cversion varchar(3) not null,
Date date not null,
foreign key(Cname, Cversion) references COMPONENT(Cname, Cversion) on delete cascade on update cascade,
foreign key(Date) references INSPECTION(Date) on delete cascade on update cascade
);

create table CONDUCT
(
ID integer not null,
Date date not null,
foreign key(ID) references EMPLOYEE(ID) on delete cascade on update cascade,
foreign key(Date) references INSPECTION(Date) on delete cascade on update cascade
);
```

Part 2: Insert

```
insert into SOFTWARE_PRODUCT (Pname,Pversion) values ('Excel', '2010');
insert into SOFTWARE_PRODUCT (Pname,Pversion) values ('Excel', '2015');
insert into SOFTWARE_PRODUCT (Pname,Pversion) values ('Excel', '2018beta');
insert into SOFTWARE_PRODUCT (Pname,Pversion) values ('Excel', 'secret');

insert into COMPONENT (Cname, Cversion, Language, Size) values ('Keyboard Driver', 'K11', 'C', 1200);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Touch Screen Driver', 'T00', 'C++', 4000);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Dbase Interface', 'D00', 'C++', 2500);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Dbase Interface', 'D01', 'C++', 2500);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Chart generator', 'C11', 'Java', 6500);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Pen driver', 'P01', 'C', 3575);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Math unit', 'A01', 'C', 5000);
insert into COMPONENT (Cname, Cversion, Language, Size) values ('Math unit', 'A02', 'Java', 3500);

insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10100, 'Employee-1', '1984-11-08', NULL);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10200, 'Employee-2', '1994-11-08', 10100);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10300, 'Employee-3', '2004-11-08', 10200);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10400, 'Employee-4', '2008-11-01', 10200);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10500, 'Employee-5', '2015-11-01', 10400);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10600, 'Employee-6', '2015-11-01', 10400);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10700, 'Employee-7', '2016-11-01', 10400);
insert into EMPLOYEE (ID, Ename, Hiredate, Mgr) values (10800, 'Employee-8', '2017-11-01', 10200);


insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Keyboard Driver', 'K11', '2010-02-14', 100, 'legacy code which is already approved');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Touch Screen Driver', 'T00', '2017-06-01', 95, 'initial release ready for usage');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Dbase Interface', 'D00', '2010-02-22', 55, 'too many hard coded parameters, the software must be more maintainable and configurable bec
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Dbase Interface', 'D00', '2010-02-24', 78, 'improved, but only handles DB2 format');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Dbase Interface', 'D00', '2010-02-26', 95, 'Okay, handles DB3 format');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Dbase Interface', 'D00', '2010-02-28', 100, 'satisified');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Dbase Interface', 'D01', '2011-05-01', 100, 'Okay ready for use');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Pen driver', 'P01', '2017-07-15', 80, 'Okay ready for beta testing');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Math unit', 'A01', '2014-06-10', 90, 'almost ready');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Math unit', 'A02', '2014-06-15', 70, 'Accuracy problems!');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Math unit', 'A02', '2014-06-30', 100, 'Okay problems fixed');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Math unit', 'A02', '2016-11-02', 100, 're-review for new employee to gain experience in the process');
```

```
insert into HAVE values ('Excel', '2010', 'Keyboard Driver', 'K11');
insert into HAVE values ('Excel', '2010', 'Dbase Interface', 'D00');
insert into HAVE values ('Excel', '2015', 'Keyboard Driver', 'K11');
insert into HAVE values ('Excel', '2015', 'Dbase Interface', 'D01');
insert into HAVE values ('Excel', '2015', 'Pen driver', 'P01');
insert into HAVE values ('Excel', '2018beta', 'Keyboard Driver', 'K11');
insert into HAVE values ('Excel', '2018beta', 'Touch Screen Driver', 'T00');
insert into HAVE values ('Excel', '2018beta', 'Chart generator', 'C11');
insert into HAVE values ('Excel', 'secret', 'Keyboard Driver', 'K11');
insert into HAVE values ('Excel', 'secret', 'Touch Screen Driver', 'T00');
insert into HAVE values ('Excel', 'secret', 'Chart generator', 'C11');
insert into HAVE values ('Excel', 'secret', 'Math unit', 'A02');


insert into OWN values ('Keyboard Driver', 'K11', 10100);
insert into OWN values ('Touch Screen Driver', 'T00', 10100);
insert into OWN values ('Dbase Interface', 'D00', 10200);
insert into OWN values ('Dbase Interface', 'D01', 10300);
insert into OWN values ('Chart generator', 'C11', 10200);
insert into OWN values ('Pen driver', 'P01', 10700);
insert into OWN values ('Math unit', 'A01', 10200);
insert into OWN values ('Math unit', 'A02', 10200);

insert into INSPECTED values ('Keyboard Driver', 'K11', '2010-02-14');
insert into INSPECTED values ('Touch Screen Driver', 'T00', '2017-06-01');
insert into INSPECTED values ('Dbase Interface', 'D00', '2010-02-22');
insert into INSPECTED values ('Dbase Interface', 'D00', '2010-02-24');
insert into INSPECTED values ('Dbase Interface', 'D00', '2010-02-26');
insert into INSPECTED values ('Dbase Interface', 'D00', '2010-02-28');
insert into INSPECTED values ('Dbase Interface', 'D01', '2011-05-01');
insert into INSPECTED values ('Pen driver', 'P01', '2017-07-15');
insert into INSPECTED values ('Math unit', 'A01', '2014-06-10');
insert into INSPECTED values ('Math unit', 'A02', '2014-06-15');
insert into INSPECTED values ('Math unit', 'A02', '2014-06-30');
insert into INSPECTED values ('Math unit', 'A02', '2016-11-02');

insert into CONDUCT values (10100, '2010-02-14');
insert into CONDUCT values (10200, '2017-06-01');
insert into CONDUCT values (10100, '2010-02-22');
insert into CONDUCT values (10100, '2010-02-24');
insert into CONDUCT values (10100, '2010-02-26');
insert into CONDUCT values (10100, '2010-02-28');
insert into CONDUCT values (10200, '2011-05-01');
insert into CONDUCT values (10300, '2017-06-10');
insert into CONDUCT values (10100, '2014-06-10');
insert into CONDUCT values (10100, '2014-06-15');
insert into CONDUCT values (10100, '2014-06-30');
insert into CONDUCT values (10700, '2016-11-02');
```

Part 3: Create Triggers

(1) Trigger for EMPLOYEE INSERT to consider the SENIORITY of the employee:

```
delimiter //
create trigger EMPLOYEE_INS before insert on EMPLOYEE for each row
begin
    if(exists(select * from employee where ID = new.Mgr) or new.ID = 10100) then
        if (new.hiredate > date_sub(now(), interval 1 year)) then
            set new.seniority = "newbie";
        else
            if (new.hiredate > date_sub(now(), interval 5 year)) then
                set new.seniority = "junior";
            else
                set new.seniority = "senior";
            end if;
        end if;
    else
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Mgr data is wrong";
    end if;
end;
//
delimiter ;
```

To Ensure this trigger can meet the requirement:

```
select * from EMPLOYEE;
```

Result Grid | Filter Rows: | Edit:

ID	Ename	Hiredate	Mgr	Seniority
10100	Employee-1	1984-11-08	NULL	senior
10200	Employee-2	1994-11-08	10100	senior
10300	Employee-3	2004-11-08	10200	senior
10400	Employee-4	2008-11-01	10200	senior
10500	Employee-5	2015-11-01	10400	junior
10600	Employee-6	2015-11-01	10400	junior
10700	Employee-7	2016-11-01	10400	junior
10800	Employee-8	2017-11-01	10200	newbie
NULL	NULL	NULL	NULL	NULL

Using select * from and the result is correct.

It can meet the requirement.

(2) Trigger for ID in EMPLOYEE to make sure ID is 5-digit number

```
delimiter //
create trigger id_INS before insert on EMPLOYEE for each row
begin
    if (LENGTH(new.ID) != 5) then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "ID should be 5 digit number";
    end if;
end
//
delimiter ;
```

To Ensure this trigger can meet the requirement:

```
insert into EMPLOYEE(ID, Ename, Hiredate, Mgr) values (100000, 'Employee-not exist', '1994-02-21', 10100);
```

Try to insert a row with length of ID not equal to 5,

```
400  05:14:41  insert into EMPLOYEE(ID, Ename, Hiredate, Mgr) values (100000, 'Employee-not exist', '1994-02-21', 10100)    Error Code: 1644. ID should be 5 digit number
```

Meet the requirement

(3) Trigger for Score in INSPECTION to make sure score is 0-100

```
delimiter //
create trigger score_INS before insert on INSPECTION for each row
begin
    if new.Score > 100 or new.Score < 0 then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Score should be a value between 0 and 100 or null";
    end if;
end
//
delimiter ;
```

To Ensure this trigger can meet the requirement:

```
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Keyboard Driver', 'K11', '2010-02-14', 120, 'incorrect score');
```

Try to insert a row with score not in 0-100.

```
401  05:18:28  insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Keyboard Driver', 'K11', '2010-02-14', 120, 'incorrect score')    Error Code: 1644. Score should be a value between 0 and 100 or null
```

Meet the requirement.

(4) Trigger for Score in INSPECTION cannot be changed

```
delimiter //
create trigger score_UPD before update on INSPECTION for each row
begin
    if new.Score is not null then
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = "Score can never be changed";
    end if;
end
//
delimiter ;
```

To Ensure this trigger can meet the requirement:

```
update INSPECTION
set score = 99
where Cname = 'Keyboard Driver' and Cversion = 'K11';
```

Try to change the score in INSPECTION:

```
402  05:20:36  update INSPECTION set score = 99 where Cname = 'Keyboard Driver' and Cversion = 'K11'    Error Code: 1644. Score can never be changed
```

Meet the requirement.

(5) Trigger for cstatus in COMPONENT to consider the status of COMPONENT, it should be only "ready", "not-ready", or "usable".

```
delimiter //
create trigger cstatus_INS after insert on INSPECTION for each row
begin
    if new.score > 90 then
        update COMPONENT, INSPECTION
        set COMPONENT.Cstatus = 'ready'
        where COMPONENT.Cname = new.Cname and COMPONENT.Cversion = new.Cversion
        and new.Date = (select max(Date) from INSPECTION where
        INSPECTION.Cname = new.Cname and INSPECTION.Cversion = new.Cversion);
    elseif new.score < 75 then
        update COMPONENT, INSPECTION
        set COMPONENT.Cstatus = 'not-ready'
        where COMPONENT.Cname = new.Cname and COMPONENT.Cversion = new.Cversion
        and new.Date = (select max(Date) from INSPECTION where
        INSPECTION.Cname = new.Cname and INSPECTION.Cversion = new.Cversion);
    else
        update COMPONENT, INSPECTION
        set COMPONENT.Cstatus = 'usable'
        where COMPONENT.Cname = new.Cname and COMPONENT.Cversion = new.Cversion
        and new.Date = (select max(Date) from INSPECTION where
        INSPECTION.Cname = new.Cname and INSPECTION.Cversion = new.Cversion);
    end if;
end
//
delimiter ;
```

When score is larger than 90, it should be 'ready', if score is less than 75, it should be 'not-ready', otherwise, it should be 'usable'.

This trigger can choose the newest status of component because of the query "select max(Date)…" which means it can choose the latest data and take it as the newest status.

For example:

```
select * from COMPONENT;
```

Result Grid | Filter Rows: | Edit:

Cname	Cversion	Language	Size	Cstatus
Chart generator	C11	Java	6500	not-ready
Dbase Interface	D00	C++	2500	not-ready
Dbase Interface	D01	C++	2500	not-ready
Keyboard Driver	K11	C	1200	not-ready
Math unit	A01	C	5000	not-ready
Math unit	A02	Java	3500	not-ready
Pen driver	P01	C	3575	not-ready
Touch Screen Driver	T00	C++	4000	not-ready
NULL	NULL	NULL	NULL	NULL

This is the initial statement.

```
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Keyboard Driver', 'K11', '2010-02-14', 100, 'legacy code which is already approved');
insert into INSPECTION (Cname, Cversion, Date, Score, Description) values ('Touch Screen Driver', 'T00', '2017-06-01', 95, 'initial release ready for usage');
```

INSERT some values into INSPECTION, awake the trigger.

```
select * from COMPONENT;
```

Result Grid | Filter Rows: | Edit:

Cname	Cversion	Language	Size	Cstatus
Chart generator	C11	Java	6500	not-ready
Dbase Interface	D00	C++	2500	not-ready
Dbase Interface	D01	C++	2500	not-ready
Keyboard Driver	K11	C	1200	ready
Math unit	A01	C	5000	not-ready
Math unit	A02	Java	3500	not-ready
Pen driver	P01	C	3575	not-ready
Touch Screen Driver	T00	C++	4000	ready
NULL	NULL	NULL	NULL	NULL

From the capture we can see that because of the insertion of ‘Keyboard Driver’ and ‘Touch Screen Driver’. The value in the COMPONENT also changed, which means the trigger works correctly.

It meets the requirement.

(6) Trigger for pstatus in SOFTWARE_PRODUCT to consider the status of SOFTWARE_PRODUCT, it also should be “ready”, “not-ready” or “usable”.

It should be judged as the worst status among its COMPONENT, so should

build the relation between the SOFTWARE_PRODUCT and COMPONENT, then create a trigger to consider the status of SOFTWARE_PRODUCT.

Create two triggers for Pstatus.

```
delimiter //
create trigger pstatus_INS after update on COMPONENT for each row
begin
    set @Pname = (select Pname from HAVE
    where Cname = new.Cname and Cversion = new.Cversion);
    set @Pversion = (select Pversion from HAVE
    where Cname = new.Cname and Cversion = new.Cversion);
    set @Cstatus = new.Cstatus;
    set @Pstatus = (select Pstatus from SOFTWARE_PRODUCT
    where SOFTWARE_PRODUCT.Pname = @Pname and SOFTWARE_PRODUCT.Pversion = @Pversion);
    update SOFTWARE_PRODUCT
        set Pstatus = @Cstatus
    where (SOFTWARE_PRODUCT.Pname = @Pname and SOFTWARE_PRODUCT.Pversion = @Pversion);
end
//
delimiter ;
```

```
delimiter //
create trigger pstatus_UPD after update on SOFTWARE_PRODUCT for each row
begin
    declare i int default 0;
    set @Pname = (select Pname from HAVE
    where Cname = new.Cname and Cversion = new.Cversion);
    set @Pversion = (select Pversion from HAVE
    where Cname = new.Cname and Cversion = new.Cversion);
    set @loopn = (select count(*) from HAVE
    where HAVE.Pname = Pname and HAVE.Pversion = Pversion);
    set @Pstatus = (select Pstatus from SOFTWARE_PRODUCT
    where SOFTWARE_PRODUCT.Pname = @Pname and SOFTWARE_PRODUCT.Pversion = @Pversion);
    get_loop : LOOP
        if @loop = i then
            LEAVE get_loop;
        end if;
    set @Cstatus = (select COMPONENT.Cstatus from COMPONENT
    where (COMPONENT.Cname = (select HAVE.Cname from HAVE
    where HAVE.Pname = @Pname and HAVE.Pversion = @Pversion limit i,1) and
    (select HAVE.Cversion from HAVE
    where HAVE.Pname = @Pname and HAVE.Pversion = @Pversion limit i,1)));
    if @Cstatus = 'not-ready' then
        set @Pstatus = 'not-ready';
    elseif @Pstatus = 'ready' and @Cstatus = 'usable' then
        set @pstatus = 'usable';
    else
        set @pstatus = 'not-ready';
    end if;
        set i = i + 1;
    end LOOP;
    update SOFTWARE_PRODUCT set SOFTWARE_PRODUCT.Pstatus = @pstatus
    where SOFTWARE_PRODUCT.Pname = @Pname and SOFTWARE_PRODUCT.Pversion = @Pversion;
end;
//
delimiter ;
```

Part 4: Select * from and Describe

(1) SOFTWARE_PRODUCT

SELECT:

```
select * from SOFTWARE_PRODUCT;
```

Result Grid | Filter Rows: | Edit:

Pname	Pversion	Pstatus
Excel	2010	readv
Excel	2015	usable
Excel	2018beta	not-readv
Excel	secret	not-readv
NULL	NULL	NULL

DESCRIBE:

```
Describe SOFTWARE_PRODUCT;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Pname	varchar(20)	NO	PRI	NULL	
Pversion	varchar(20)	NO	PRI	NULL	
Pstatus	enum('readv','usable','not-readv')	YES		not-readv	

(2) COMPONENT

SELECT:

```
select * from COMPONENT;
```

Result Grid | Filter Rows: | Edit:

Cname	Cversion	Language	Size	Cstatus
Chart generator	C11	Java	6500	not-readv
Dbase Interface	D00	C++	2500	readv
Dbase Interface	D01	C++	2500	readv
Keyboard Driver	K11	C	1200	readv
Math unit	A01	C	5000	usable
Math unit	A02	Java	3500	readv
Pen driver	P01	C	3575	usable
Touch Screen Driver	T00	C++	4000	readv
NULL	NULL	NULL	NULL	NULL

DESCRIBE:

```
Describe COMPONENT;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Field	Type	Null	Key	Default	Extra
Cname	varchar(50)	NO	PRI	NULL	
Cversion	varchar(3)	NO	PRI	NULL	
Language	enum('C','C++','C#','Java','PHP')	YES		NULL	
Size	int(11)	NO		NULL	
Cstatus	enum('ready','usable','not-ready')	YES		not-ready	

(3) EMPLOYEE

SELECT:

```
select * from EMPLOYEE;
```

Result Grid | Filter Rows: | Edit:

ID	Ename	Hiredate	Mgr	Seniority
10100	Employee-1	1984-11-08	NULL	senior
10200	Employee-2	1994-11-08	10100	senior
10300	Employee-3	2004-11-08	10200	senior
10400	Employee-4	2008-11-01	10200	senior
10500	Employee-5	2015-11-01	10400	junior
10600	Employee-6	2015-11-01	10400	junior
10700	Employee-7	2016-11-01	10400	junior
10800	Employee-8	2017-11-01	10200	newbie
NULL	NULL	NULL	NULL	NULL

DESCRIBE:

```
Describe EMPLOYEE;
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	PRI	NULL	
Ename	varchar(60)	NO		NULL	
Hiredate	date	NO		NULL	
Mgr	int(11)	YES		NULL	
Seniority	varchar(10)	YES		NULL	

(4) INSPECTION

SELECT:

```
select * from INSPECTION;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap C

Cname	Cversion	Date	Score	Description
Keyboard Driver	K11	2010-02-14	100	legacy code which is already approved
Touch Screen Driver	T00	2017-06-01	95	initial release ready for usage
Dbase Interface	D00	2010-02-22	55	too many hard coded parameters, the software...
Dbase Interface	D00	2010-02-24	78	improved, but only handles DB2 format
Dbase Interface	D00	2010-02-26	95	Okay, handles DB3 format
Dbase Interface	D00	2010-02-28	100	satisified
Dbase Interface	D01	2011-05-01	100	Okay ready for use
Pen driver	P01	2017-07-15	80	Okay ready for beta testing
Math unit	A01	2014-06-10	90	almost ready
Math unit	A02	2014-06-15	70	Accuracy problems!
Math unit	A02	2014-06-30	100	Okay problems fixed
Math unit	A02	2016-11-02	100	re-review for new employee to gain experience ...
NULL	NULL	NULL	NULL	NULL

DESCRIBE:

```
Describe INSPECTION;
```

Result Grid | Filter Rows: | Export: | Wrap

Field	Type	Null	Key	Default	Extra
Cname	varchar(50)	NO	MUL	NULL	
Cversion	varchar(3)	NO		NULL	
Date	date	NO	MUL	NULL	
Score	int(11)	YES		NULL	
Description	varchar(4000)	NO		NULL	

(5) HAVE

SELECT:

```
select * from HAVE;
```

Result Grid | Filter Rows: | Edit:

Pname	Pversion	Cname	Cversion
Excel	2010	Keyboard Driver	K11
Excel	2010	Dbase Interface	D00
Excel	2015	Keyboard Driver	K11
Excel	2015	Dbase Interface	D01
Excel	2015	Pen driver	P01
Excel	2018beta	Keyboard Driver	K11
Excel	2018beta	Touch Screen Driver	T00
Excel	2018beta	Chart generator	C11
Excel	secret	Keyboard Driver	K11
Excel	secret	Touch Screen Driver	T00
Excel	secret	Chart generator	C11
Excel	secret	Math unit	A02
NULL	NULL	NULL	NULL

DESCRIBE:

```
Describe HAVE;
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
Pname	varchar(10)	NO	MUL	NULL	
Pversion	varchar(10)	NO		NULL	
Cname	varchar(50)	NO	MUL	NULL	
Cversion	varchar(3)	NO		NULL	

(6) OWN

SELECT:

```
select * from OWN;
```

Result Grid | Filter Rows:

Cname	Cversion	ID
Keyboard Driver	K11	10100
Touch Screen Driver	T00	10100
Dbase Interface	D00	10200
Dbase Interface	D01	10300
Chart generator	C11	10200
Pen driver	P01	10700
Math unit	A01	10200
Math unit	A02	10200
NULL	NULL	NULL

DESCRIBE:

```
Describe OWN;
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
Cname	varchar(50)	NO	MUL	NULL	
Cversion	varchar(3)	NO		NULL	
ID	int(11)	NO	MUL	NULL	

(7) INSPECTED

SELECT:

```
select * from INSPECTED;
```

Result Grid | Filter Rows:

Cname	Cversion	Date
Keyboard Driver	K11	2010-02-14
Touch Screen Driver	T00	2017-06-01
Dbase Interface	D00	2010-02-22
Dbase Interface	D00	2010-02-24
Dbase Interface	D00	2010-02-26
Dbase Interface	D00	2010-02-28
Dbase Interface	D01	2011-05-01
Pen driver	P01	2017-07-15
Math unit	A01	2014-06-10
Math unit	A02	2014-06-15
Math unit	A02	2014-06-30
Math unit	A02	2016-11-02
NULL	NULL	NULL

DESCRIBE:

```
DESCRIBE INSPECTED;
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
Cname	varchar(50)	NO	MUL	NULL	
Cversion	varchar(3)	NO		NULL	
Date	date	NO	MUL	NULL	

(8) CONDUCT

SELECT:

```
select * from CONDUCT;
```

Result Grid | Filter Rows:

ID	Date
10100	2010-02-14
10200	2017-06-01
10100	2010-02-22
10100	2010-02-24
10100	2010-02-26
10100	2010-02-28
10200	2011-05-01
10300	2017-07-15
10100	2014-06-10
10100	2014-06-15
10100	2014-06-30
10700	2016-11-02
NULL	NULL

DESCRIBE:

```
DESCRIBE CONDUCT;
```

Result Grid | Filter Rows: | Export:

Field	Type	Null	Key	Default	Extra
ID	int(11)	NO	MUL	NULL	
Date	date	NO	MUL	NULL	

Part 5: Explanation

In phase 1, I give an attribute in the relational table "HAVE" called "SOFTWARE_BUILD" while in phase 2 I choose not to use it because I found that I no longer need this attribute.

In addition, because of the values the phase 2 gave me, I have to add two attributes in table "EMPLOYEE", one is "Mgr" which store the manager id, another is "hiredate" which is the important attribute that can consider the SENIORITY of the employee.

I think I can improve or create more triggers to meet the requirement.