

Database implementation and Testing

Database Implementation

Note: Although data can be removed from certain tables, it might still be relevant to keep the data of related objects. For that reason, some NOT NULL constraints included in the submission of part 1 were removed. This way, ON DELETE SET NULL operations could be used. One example is the relationship between Problem and Equipment -- If the equipment gets deleted from the database, it is still important that the Problem data remains in the database (instead of being cascaded) as a way to retrieve relevant data for future reports.

The SQL statements for task 3 are included below. You can also download the **.sql** file if you want to see a prettier version on your text editor.

File: [create_db_help_desk.sql](#)

PROMPT 'Dropping tables for a clean initialisation'

```
DROP TABLE SOFTWARE_LICENSE_STATUS CASCADE CONSTRAINTS;
DROP TABLE SOFTWARE_LICENSE CASCADE CONSTRAINTS;
DROP TABLE SOFTWARE CASCADE CONSTRAINTS;
DROP TABLE SOFTWARE_VERSION CASCADE CONSTRAINTS;
DROP TABLE EQUIPMENT_SOFTWARE CASCADE CONSTRAINTS;
DROP TABLE OPERATING_SYSTEM CASCADE CONSTRAINTS;
DROP TABLE OPERATING_SYSTEM_VERSION CASCADE CONSTRAINTS;
DROP TABLE EQUIPMENT_OPERATING_SYSTEM CASCADE CONSTRAINTS;
DROP TABLE EQUIPMENT CASCADE CONSTRAINTS;
DROP TABLE EQUIPMENT_MODEL CASCADE CONSTRAINTS;
DROP TABLE EQUIPMENT_TYPE CASCADE CONSTRAINTS;
DROP TABLE EQUIPMENT_MAKE CASCADE CONSTRAINTS;
DROP TABLE PROBLEM_TYPE_SPECIALIST CASCADE CONSTRAINTS;
DROP TABLE PROBLEM_TYPE CASCADE CONSTRAINTS;
DROP TABLE PROBLEM_ASSIGNMENT CASCADE CONSTRAINTS;
DROP TABLE PROBLEM_STATUS CASCADE CONSTRAINTS;
DROP TABLE PROBLEM_SOLUTION CASCADE CONSTRAINTS;
DROP TABLE PROBLEM CASCADE CONSTRAINTS;
DROP TABLE PROBLEM_CALL CASCADE CONSTRAINTS;
DROP TABLE CALL CASCADE CONSTRAINTS;
DROP TABLE STAFF CASCADE CONSTRAINTS;
DROP TABLE STAFF_ROLE CASCADE CONSTRAINTS;
DROP TABLE CALL_STATUS CASCADE CONSTRAINTS;
DROP TABLE CALLER_PHONE CASCADE CONSTRAINTS;
DROP TABLE CALLER CASCADE CONSTRAINTS;
```

```
DROP SEQUENCE CALLER_PHONE_ID_SEQ;
DROP SEQUENCE CALL_STATUS_ID_SEQ;
DROP SEQUENCE STAFF_ROLE_ID_SEQ;
DROP SEQUENCE STAFF_ID_SEQ;
DROP SEQUENCE CALL_ID_SEQ;
DROP SEQUENCE PROBLEM_CALL_ID_SEQ;
DROP SEQUENCE PROBLEM_ID_SEQ;
DROP SEQUENCE PROBLEM_SOLUTION_ID_SEQ;
DROP SEQUENCE PROBLEM_STATUS_ID_SEQ;
DROP SEQUENCE PROBLEM_ASSIGNMENT_ID_SEQ;
DROP SEQUENCE PROBLEM_TYPE_ID_SEQ;
DROP SEQUENCE PROBLEM_TYPE_SPECIALIST_ID_SEQ;
DROP SEQUENCE EQUIPMENT_ID_SEQ;
DROP SEQUENCE EQUIPMENT_MODEL_ID_SEQ;
DROP SEQUENCE EQUIPMENT_MAKE_ID_SEQ;
DROP SEQUENCE EQUIPMENT_TYPE_ID_SEQ;
DROP SEQUENCE EQUIPMENT_OPERATING_SYSTEM_ID_SEQ;
DROP SEQUENCE OPERATING_SYSTEM_VERSION_ID_SEQ;
DROP SEQUENCE OPERATING_SYSTEM_ID_SEQ;
DROP SEQUENCE EQUIPMENT_SOFTWARE_ID_SEQ;
DROP SEQUENCE SOFTWARE_VERSION_ID_SEQ;
DROP SEQUENCE SOFTWARE_ID_SEQ;
DROP SEQUENCE SOFTWARE_LICENSE_ID_SEQ;
DROP SEQUENCE SOFTWARE_LICENSE_STATUS_ID_SEQ;
```

```
REM === INITIALISING TABLES ===
```

```
PROMPT 'Creating tables'
```

```
CREATE TABLE CALLER
(ID CHAR(10)
,FIRST_NAME VARCHAR2(30)
,LAST_NAME VARCHAR2(30)
)
/
CREATE TABLE CALLER_PHONE
(ID NUMBER(8,0)
,PHONE_NUMBER CHAR(12)
,CALLER_ID CHAR(10)
)
/
CREATE TABLE CALL_STATUS
(ID NUMBER(8,0)
```

```

,NAME VARCHAR2(30)
)
/
CREATE TABLE STAFF_ROLE
(ID NUMBER(8,0)
,NAME VARCHAR2(30)
)
/
CREATE TABLE STAFF
(ID NUMBER(8,0)
,FIRST_NAME VARCHAR2(30)
,LAST_NAME VARCHAR2(30)
,STAFF_ROLE_ID NUMBER(8,0)
)
/
CREATE TABLE CALL
(ID NUMBER(8,0)
,CALLER_ID CHAR(10)
,CALL_STATUS_ID NUMBER(8,0)
,STAFF_ID NUMBER(8,0)
,DATETIME_STARTED TIMESTAMP(2)
,DATETIME_FINISHED TIMESTAMP(2)
)
/
CREATE TABLE PROBLEM_CALL
(ID NUMBER(8,0)
,CALL_ID NUMBER(8,0)
,PROBLEM_ID NUMBER(8,0)
)
/
CREATE TABLE PROBLEM
(ID NUMBER(8,0)
,PROBLEM_STATUS_ID NUMBER(8,0)
,PROBLEM_TYPE_ID NUMBER(8,0)
,EQUIPMENT_ID NUMBER(8,0)
,DESCRIPTION VARCHAR2(2000)
,DATETIME_CREATED TIMESTAMP(2)
)
/
CREATE TABLE PROBLEM_SOLUTION
(ID NUMBER(8,0)
,PROBLEM_ID NUMBER(8,0)
,DESCRIPTION VARCHAR2(2000)
,DATETIME_CREATED TIMESTAMP(2)

```

```

)
/
CREATE TABLE PROBLEM_STATUS
(ID NUMBER(8,0)
,NAME VARCHAR2(30)
)
/
CREATE TABLE PROBLEM_ASSIGNMENT
(ID NUMBER(8,0)
,PROBLEM_ID NUMBER(8,0)
,STAFF_ID NUMBER(8,0)
,DATETIME_CREATED TIMESTAMP(2)
)
/
CREATE TABLE PROBLEM_TYPE
(ID NUMBER(8,0)
,NAME VARCHAR2(30)
)
/
CREATE TABLE PROBLEM_TYPE_SPECIALIST
(ID NUMBER(8,0)
,PROBLEM_TYPE_ID NUMBER(8,0)
,STAFF_ID NUMBER(8,0)
)
/
CREATE TABLE EQUIPMENT_MAKE
(ID NUMBER(8,0)
,NAME VARCHAR2(30)
)
/
CREATE TABLE EQUIPMENT_TYPE
(ID NUMBER(8,0)
,NAME VARCHAR2(30)
)
/
CREATE TABLE EQUIPMENT_MODEL
(ID NUMBER(8,0)
,NAME VARCHAR2(100)
,RELEASE_DATE DATE
,EQUIPMENT_TYPE_ID NUMBER(8,0)
,EQUIPMENT_MAKE_ID NUMBER(8,0)
)
/
CREATE TABLE EQUIPMENT

```

```

(ID NUMBER(8,0)
,IDENTIFIER VARCHAR2(200)
,EQUIPMENT_MODEL_ID NUMBER(8,0)
)
/
CREATE TABLE EQUIPMENT_OPERATING_SYSTEM
(ID NUMBER(8,0)
,EQUIPMENT_ID NUMBER(8,0)
,OPERATING_SYSTEM_VERSION_ID NUMBER(8,0)
)
/
CREATE TABLE OPERATING_SYSTEM_VERSION
(ID NUMBER(8,0)
,NAME VARCHAR2(100)
,RELEASE_DATE DATE
,OPERATING_SYSTEM_ID NUMBER(8,0)
)
/
CREATE TABLE OPERATING_SYSTEM
(ID NUMBER(8,0)
,NAME VARCHAR2(100)
)
/
CREATE TABLE EQUIPMENT_SOFTWARE
(ID NUMBER(8,0)
,EQUIPMENT_ID NUMBER(8,0)
,SOFTWARE_VERSION_ID NUMBER(8,0)
)
/
CREATE TABLE SOFTWARE_VERSION
(ID NUMBER(8,0)
,NAME VARCHAR2(100)
,RELEASE_DATE DATE
,SOFTWARE_ID NUMBER(8,0)
)
/
CREATE TABLE SOFTWARE
(ID NUMBER(8,0)
,NAME VARCHAR2(100)
)
/
CREATE TABLE SOFTWARE_LICENSE
(ID NUMBER(8,0)
,IDENTIFIER VARCHAR2(1000)

```

```
,SOFTWARE_VERSION_ID NUMBER(8,0)
)
/
```

```
CREATE TABLE SOFTWARE_LICENSE_STATUS
(ID NUMBER(8,0)
,IS_VALID CHAR(1)
,SOFTWARE_LICENSE_ID NUMBER(8,0)
)
/
```

```
REM === INITIALISING INDEXES ===
PROMPT 'Creating Indexes'
```

```
CREATE INDEX CALLR_PHON_CALLR_FK_I ON CALLER_PHONE(CALLER_ID)
/
CREATE INDEX STF_STF_ROL_FK_I ON STAFF(STAFF_ROLE_ID)
/
CREATE INDEX CALL_CALLR_FK_I ON CALL(CALLER_ID)
/
CREATE INDEX CALL_CALL_STA_FK_I ON CALL(CALL_STATUS_ID)
/
CREATE INDEX CALL_STF_FK_I ON CALL(STAFF_ID)
/
CREATE INDEX PROB_CALL_CALL_FK_I ON PROBLEM_CALL(CALL_ID)
/
CREATE INDEX PROB_CALL_PROB_FK_I ON PROBLEM_CALL(PROBLEM_ID)
/
CREATE INDEX PROB_PROB_STA_FK_I ON PROBLEM(PROBLEM_STATUS_ID)
/
CREATE INDEX PROB_PROB_TYP_FK_I ON PROBLEM(PROBLEM_TYPE_ID)
/
CREATE INDEX PROB_EQP_FK_I ON PROBLEM(EQUIPMENT_ID)
/
CREATE INDEX PROB_SOL_PROB_FK_I ON PROBLEM_SOLUTION(PROBLEM_ID)
/
CREATE INDEX PROB_ASG_PROB_FK_I ON PROBLEM_ASSIGNMENT(PROBLEM_ID)
/
CREATE INDEX PROB_ASG_STF_FK_I ON PROBLEM_ASSIGNMENT(STAFF_ID)
/
CREATE INDEX PROB_TYP_SPEC_PROB_TYP_FK_I
ON PROBLEM_TYPE_SPECIALIST(PROBLEM_TYPE_ID)
/
CREATE INDEX PROB_TYP_SPEC_STF_FK_I ON
PROBLEM_TYPE_SPECIALIST(STAFF_ID)
```

```

/
CREATE INDEX EQP_EQP_MDL_FK_I ON EQUIPMENT(EQUIPMENT_MODEL_ID)
/
CREATE INDEX EQP_MDL_EQP_TYP_FK_I ON
EQUIPMENT_MODEL(EQUIPMENT_TYPE_ID)
/
CREATE INDEX EQP_MDL_EQP_MK_FK_I ON
EQUIPMENT_MODEL(EQUIPMENT_MAKE_ID)
/
CREATE INDEX EQP_OS_EQP_FK_I ON
EQUIPMENT_OPERATING_SYSTEM(EQUIPMENT_ID)
/
CREATE INDEX EQP_OS_OS_VER_PK_I
ON EQUIPMENT_OPERATING_SYSTEM(OPERATING_SYSTEM_VERSION_ID)
/
CREATE INDEX OS_VER_OS_FK ON
OPERATING_SYSTEM_VERSION(OPERATING_SYSTEM_ID)
/
CREATE INDEX EQP_SFT_EQP_FK_I ON EQUIPMENT_SOFTWARE(EQUIPMENT_ID)
/
CREATE INDEX EQP_SFT_SFT_VER_PK_I ON
EQUIPMENT_SOFTWARE(SOFTWARE_VERSION_ID)
/
CREATE INDEX SFT_VER_SFT_FK ON SOFTWARE_VERSION(SOFTWARE_ID)
/
CREATE INDEX SFT_LIC_SFT_VER_FK_I ON
SOFTWARE_LICENSE(SOFTWARE_VERSION_ID)
/
CREATE INDEX SFT_LIC_STA_SFT_LIC_FK_I
ON SOFTWARE_LICENSE_STATUS(SOFTWARE_LICENSE_ID)
/

```

```

COMMIT;

```

```

REM === INITIALISING CHECK CONSTRAINTS ===

```

```

PROMPT 'Creating NOT NULL constraints'

```

```

ALTER TABLE CALLER
MODIFY(ID CONSTRAINT CLLR_ID_NNULL NOT NULL)
MODIFY(LAST_NAME CONSTRAINT CLLR_LAST_NAME_NNULL NOT NULL)
/
ALTER TABLE CALLER_PHONE
MODIFY(ID CONSTRAINT CLLR_PHON_ID_NNULL NOT NULL)

```

```

MODIFY(PHONE_NUMBER CONSTRAINT CLLR_PHON_NUM_NNULL NOT NULL)
MODIFY(CALLER_ID CONSTRAINT CLLR_PHON_CALLR_ID_NNULL NOT NULL)
/
ALTER TABLE CALL_STATUS
MODIFY(ID CONSTRAINT CALL_STAT_ID_NNULL NOT NULL)
MODIFY(NAME CONSTRAINT CALL_STAT_NAME_NNULL NOT NULL)
/
ALTER TABLE STAFF_ROLE
MODIFY(ID CONSTRAINT STF_ROL_ID_NNULL NOT NULL)
MODIFY(NAME CONSTRAINT STF_ROL_NAME_NNULL NOT NULL)
/
ALTER TABLE STAFF
MODIFY(ID CONSTRAINT STF_ID_NNULL NOT NULL)
MODIFY(LAST_NAME CONSTRAINT STF_LAST_NAME_NNULL NOT NULL)
/
ALTER TABLE CALL
MODIFY(ID CONSTRAINT CALL_ID_NNULL NOT NULL)
MODIFY(DATETIME_STARTED CONSTRAINT CALL_DTT_START_NNULL NOT NULL)
/
ALTER TABLE PROBLEM_CALL
MODIFY(ID CONSTRAINT PROB_CALL_ID_NNULL NOT NULL)
MODIFY(CALL_ID CONSTRAINT PROB_CALL_CALL_NNULL NOT NULL)
MODIFY(PROBLEM_ID CONSTRAINT PROB_CALL_PROB_ID_NNULL NOT NULL)
/
ALTER TABLE PROBLEM
MODIFY(ID CONSTRAINT PROB_ID_NNULL NOT NULL)
MODIFY(DESCRIPTION CONSTRAINT PROB_DESCR_NNULL NOT NULL)
MODIFY(DATETIME_CREATED CONSTRAINT PROB_DT_CRE_NNULL NOT NULL)
/
ALTER TABLE PROBLEM_SOLUTION
MODIFY(ID CONSTRAINT PROB_SOL_ID_NNULL NOT NULL)
MODIFY(PROBLEM_ID CONSTRAINT PROB_SOL_PROB_NNULL NOT NULL)
MODIFY(DESCRIPTION CONSTRAINT PROB_SOL_DESCR_NNULL NOT NULL)
MODIFY(DATETIME_CREATED CONSTRAINT PROB_SOL_DT_CRE_NNULL NOT NULL)
/
ALTER TABLE PROBLEM_STATUS
MODIFY(ID CONSTRAINT PROB_STA_ID_NNULL NOT NULL)
MODIFY(NAME CONSTRAINT PROB_STA_NAME_NNULL NOT NULL)
/
ALTER TABLE PROBLEM_ASSIGNMENT
MODIFY(ID CONSTRAINT PROB_ASG_ID_NNULL NOT NULL)
MODIFY(PROBLEM_ID CONSTRAINT PROB_ASG_PROB_NNULL NOT NULL)
MODIFY(STAFF_ID CONSTRAINT PROB_ASG_STF_NNULL NOT NULL)
MODIFY(DATETIME_CREATED CONSTRAINT PROB_ASG_DT_CRE_NNULL NOT NULL)

```



```

/
ALTER TABLE PROBLEM_TYPE
  MODIFY(ID CONSTRAINT PROB_TYP_ID_NNULL NOT NULL)
  MODIFY(NAME CONSTRAINT PROB_TYP_NAME_NNULL NOT NULL)
/
ALTER TABLE PROBLEM_TYPE_SPECIALIST
  MODIFY(ID CONSTRAINT PROB_TYP_SPEC_NNULL NOT NULL)
  MODIFY(PROBLEM_TYPE_ID CONSTRAINT PROB_TYP_SPEC_TYP_NNULL NOT NULL)
  MODIFY(STAFF_ID CONSTRAINT PROB_TYP_SPEC_STF_NNULL NOT NULL)
/
ALTER TABLE EQUIPMENT
  MODIFY(ID CONSTRAINT EQP_ID_NNULL NOT NULL)
  MODIFY(IDENTIFIER CONSTRAINT EQP_IDT_NNULL NOT NULL)
/
ALTER TABLE EQUIPMENT_MODEL
  MODIFY(ID CONSTRAINT EQP_MDL_NNULL NOT NULL)
  MODIFY(NAME CONSTRAINT EQP_NAME_NNULL NOT NULL)
  MODIFY(RELEASE_DATE CONSTRAINT EQP_REL_DT_NNULL NOT NULL)
/
ALTER TABLE EQUIPMENT_MAKE
  MODIFY(ID CONSTRAINT EQP_MK_ID_NNULL NOT NULL)
  MODIFY(NAME CONSTRAINT EQP_MK_NAME_NNULL NOT NULL)
/
ALTER TABLE EQUIPMENT_TYPE
  MODIFY(ID CONSTRAINT EQP_TYP_ID_NNULL NOT NULL)
  MODIFY(NAME CONSTRAINT EQP_TYP_NAME_NNULL NOT NULL)
/
ALTER TABLE EQUIPMENT_OPERATING_SYSTEM
  MODIFY(ID CONSTRAINT EQP_OS_ID_NNULL NOT NULL)
  MODIFY(EQUIPMENT_ID CONSTRAINT EQP_OS_EQP_NNULL NOT NULL)
/
ALTER TABLE OPERATING_SYSTEM_VERSION
  MODIFY(ID CONSTRAINT OS_VER_ID_NNULL NOT NULL)
  MODIFY(NAME CONSTRAINT OS_VER_NAME_NNULL NOT NULL)
  MODIFY(RELEASE_DATE CONSTRAINT OS_VER_REL_DT_NNULL NOT NULL)
  MODIFY(OPERATING_SYSTEM_ID CONSTRAINT OS_VER_OS NOT NULL)
/
ALTER TABLE OPERATING_SYSTEM
  MODIFY(ID CONSTRAINT OS_ID_NNULL NOT NULL)
  MODIFY(NAME CONSTRAINT OS_NAME_NNULL NOT NULL)
/
ALTER TABLE EQUIPMENT_SOFTWARE
  MODIFY(ID CONSTRAINT EQP_SFT_ID_NNULL NOT NULL)
  MODIFY(EQUIPMENT_ID CONSTRAINT EQP_SFT_EQP_NNULL NOT NULL)

```

```

/
ALTER TABLE SOFTWARE_VERSION
MODIFY(ID CONSTRAINT SFT_VER_ID_NNULL NOT NULL)
MODIFY(NAME CONSTRAINT SFT_VER_NAME_NNULL NOT NULL)
MODIFY(RELEASE_DATE CONSTRAINT SFT_VER_REL_DT_NNULL NOT NULL)
MODIFY(SOFTWARE_ID CONSTRAINT SFT_VER_SFT NOT NULL)
/
ALTER TABLE SOFTWARE
MODIFY(ID CONSTRAINT SFT_ID_NNULL NOT NULL)
MODIFY(NAME CONSTRAINT SFT_NAME_NNULL NOT NULL)
/
ALTER TABLE SOFTWARE_LICENSE
MODIFY(ID CONSTRAINT SFT_LIC_ID_NNULL NOT NULL)
MODIFY(IDENTIFIER CONSTRAINT SFT_LIC_IDT_NNULL NOT NULL)
MODIFY(SOFTWARE_VERSION_ID CONSTRAINT SFT_LIC_SOFT_VER_NNULL NOT
NULL)
/
ALTER TABLE SOFTWARE_LICENSE_STATUS
MODIFY(ID CONSTRAINT SFT_LIC_STA_ID_NNULL NOT NULL)
MODIFY(IS_VALID CONSTRAINT SFT_LIC_VAL_NNULL NOT NULL)
MODIFY(SOFTWARE_LICENSE_ID CONSTRAINT SFT_LIC_STA_LIC_NNULL NOT NULL)
/

```

PROMPT 'Creating UNIQUE constraints'

```

ALTER TABLE CALLER_PHONE
ADD CONSTRAINT CALLR_PHON_UNIQUE UNIQUE(PHONE_NUMBER)
/
ALTER TABLE CALL_STATUS
ADD CONSTRAINT CALL_STA_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE STAFF_ROLE
ADD CONSTRAINT STF_ROL_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE PROBLEM_STATUS
ADD CONSTRAINT PROB_STA_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE PROBLEM_TYPE
ADD CONSTRAINT PROB_TYP_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE EQUIPMENT_MODEL
ADD CONSTRAINT EQP_MDL_UNIQUE
UNIQUE(EQUIPMENT_TYPE_ID, EQUIPMENT_MAKE_ID, NAME)
/

```

```

ALTER TABLE EQUIPMENT_MAKE
ADD CONSTRAINT EQP_MK_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE EQUIPMENT_TYPE
ADD CONSTRAINT EQP_TYP_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE OPERATING_SYSTEM_VERSION
ADD CONSTRAINT OS_VER_UNIQUE UNIQUE(OPERATING_SYSTEM_ID, NAME)
/
ALTER TABLE OPERATING_SYSTEM
ADD CONSTRAINT OS_NAME_UNIQUE UNIQUE(NAME)
/
ALTER TABLE SOFTWARE_VERSION
ADD CONSTRAINT SFT_VER_UNIQUE UNIQUE(NAME, SOFTWARE_ID)
/
ALTER TABLE SOFTWARE
ADD CONSTRAINT SFT_NAME_UNIQUE UNIQUE(NAME)
/

```

PROMPT 'Creating PRIMARY KEY constraints'

```

ALTER TABLE CALLER ADD CONSTRAINT CALLR_PK PRIMARY KEY (ID)
/
ALTER TABLE CALLER_PHONE ADD CONSTRAINT CALLR_PHON_PK PRIMARY KEY (ID)
/
ALTER TABLE CALL_STATUS ADD CONSTRAINT CALL_STAT_PK PRIMARY KEY (ID)
/
ALTER TABLE STAFF_ROLE ADD CONSTRAINT STF_ROL_PK PRIMARY KEY (ID)
/
ALTER TABLE STAFF ADD CONSTRAINT STF_PK PRIMARY KEY (ID)
/
ALTER TABLE CALL ADD CONSTRAINT CALL_PK PRIMARY KEY (ID)
/
ALTER TABLE PROBLEM_CALL ADD CONSTRAINT PROB_CALL_PK PRIMARY KEY (ID)
/
ALTER TABLE PROBLEM ADD CONSTRAINT PROB_PK PRIMARY KEY (ID)
/
ALTER TABLE PROBLEM_SOLUTION ADD CONSTRAINT PROB_SOL_PK PRIMARY KEY
(ID)
/
ALTER TABLE PROBLEM_STATUS ADD CONSTRAINT PROB_STA_PK PRIMARY KEY (ID)
/
ALTER TABLE PROBLEM_ASSIGNMENT ADD CONSTRAINT PROB_ASG_PK PRIMARY
KEY (ID)

```

```

/
ALTER TABLE PROBLEM_TYPE ADD CONSTRAINT PROB_TYP_PK PRIMARY KEY (ID)
/
ALTER TABLE PROBLEM_TYPE_SPECIALIST ADD CONSTRAINT
PROB_TYP_SPEC_PK PRIMARY KEY (ID)
/
ALTER TABLE EQUIPMENT ADD CONSTRAINT EQP_PK PRIMARY KEY (ID)
/
ALTER TABLE EQUIPMENT_MODEL ADD CONSTRAINT EQP_MDL_PK PRIMARY KEY (ID)
/
ALTER TABLE EQUIPMENT_MAKE ADD CONSTRAINT EQP_MK_PK PRIMARY KEY (ID)
/
ALTER TABLE EQUIPMENT_TYPE ADD CONSTRAINT EQP_TYP_PK PRIMARY KEY (ID)
/
ALTER TABLE EQUIPMENT_OPERATING_SYSTEM ADD CONSTRAINT
EQP_OS_PK PRIMARY KEY (ID)
/
ALTER TABLE OPERATING_SYSTEM_VERSION ADD CONSTRAINT OS_VER_PK
PRIMARY KEY (ID)
/
ALTER TABLE OPERATING_SYSTEM ADD CONSTRAINT OS_PK PRIMARY KEY (ID)
/
ALTER TABLE EQUIPMENT_SOFTWARE ADD CONSTRAINT EQP_SFT_PK PRIMARY KEY
(ID)
/
ALTER TABLE SOFTWARE_VERSION ADD CONSTRAINT SFT_VER_PK PRIMARY KEY (ID)
/
ALTER TABLE SOFTWARE ADD CONSTRAINT SFT_PK PRIMARY KEY (ID)
/
ALTER TABLE SOFTWARE_LICENSE ADD CONSTRAINT SFT_LIC_PK PRIMARY KEY (ID)
/
ALTER TABLE SOFTWARE_LICENSE_STATUS ADD CONSTRAINT
SFT_LIC_STA_PK PRIMARY KEY (ID)
/

```

PROMPT 'Creating FOREIGN KEY constraints'

```

ALTER TABLE CALLER_PHONE ADD CONSTRAINT
CALLR_PHON_CALLR_FK FOREIGN KEY (CALLER_ID)
REFERENCES CALLER(ID)
ON DELETE CASCADE
/
ALTER TABLE STAFF ADD CONSTRAINT
STF_STF_ROL_FK FOREIGN KEY (STAFF_ROLE_ID)

```

```

REFERENCES STAFF_ROLE(ID)
ON DELETE SET NULL
/
ALTER TABLE CALL ADD CONSTRAINT
CALL_CALLR_FK FOREIGN KEY (CALLER_ID)
REFERENCES CALLER(ID)
ON DELETE SET NULL
ADD CONSTRAINT
CALL_STF_FK FOREIGN KEY (STAFF_ID)
REFERENCES STAFF(ID)
ON DELETE SET NULL
ADD CONSTRAINT
CALL_CALL_STA_FK FOREIGN KEY (CALL_STATUS_ID)
REFERENCES CALL_STATUS(ID)
ON DELETE SET NULL
/
ALTER TABLE PROBLEM_CALL ADD CONSTRAINT
PROB_CALL_CALL_FK FOREIGN KEY (CALL_ID)
REFERENCES CALL(ID)
ON DELETE CASCADE
ADD CONSTRAINT
PROB_CALL_PROB_FK FOREIGN KEY (PROBLEM_ID)
REFERENCES PROBLEM(ID)
ON DELETE CASCADE
/
ALTER TABLE PROBLEM ADD CONSTRAINT
PROB_PROB_STA_FK FOREIGN KEY (PROBLEM_STATUS_ID)
REFERENCES PROBLEM_STATUS(ID)
ON DELETE SET NULL
ADD CONSTRAINT
PROB_PROB_TYP_FK FOREIGN KEY (PROBLEM_TYPE_ID)
REFERENCES PROBLEM_TYPE(ID)
ON DELETE SET NULL
ADD CONSTRAINT
PROB_EQP_FK FOREIGN KEY (EQUIPMENT_ID)
REFERENCES EQUIPMENT(ID)
ON DELETE SET NULL
/
ALTER TABLE PROBLEM_SOLUTION ADD CONSTRAINT
PROB_SOL_PROB_FK FOREIGN KEY (PROBLEM_ID)
REFERENCES PROBLEM(ID)
ON DELETE CASCADE
/
ALTER TABLE PROBLEM_ASSIGNMENT ADD CONSTRAINT

```

```

PROB_ASG_PROB_FK FOREIGN KEY (PROBLEM_ID)
REFERENCES PROBLEM(ID)
ON DELETE CASCADE
ADD CONSTRAINT
PROB_ASG_STF_FK FOREIGN KEY (STAFF_ID)
REFERENCES STAFF(ID)
ON DELETE CASCADE
/
ALTER TABLE PROBLEM_TYPE_SPECIALIST ADD CONSTRAINT
PROB_TYP_SPEC_PROB_TYP_FK FOREIGN KEY (PROBLEM_TYPE_ID)
REFERENCES PROBLEM_TYPE(ID)
ON DELETE CASCADE
ADD CONSTRAINT
PROB_TYP_SPEC_STF_FK FOREIGN KEY (STAFF_ID)
REFERENCES STAFF(ID)
ON DELETE CASCADE
/
ALTER TABLE EQUIPMENT ADD CONSTRAINT
EQP_EQP_MDL_FK FOREIGN KEY (EQUIPMENT_MODEL_ID)
REFERENCES EQUIPMENT_MODEL(ID)
ON DELETE SET NULL
/
ALTER TABLE EQUIPMENT_MODEL ADD CONSTRAINT
EQP_MDL_EQP_TYP_FK FOREIGN KEY (EQUIPMENT_TYPE_ID)
REFERENCES EQUIPMENT_TYPE(ID)
ON DELETE SET NULL
ADD CONSTRAINT
EQP_MDL_EQP_MK_FK FOREIGN KEY (EQUIPMENT_MAKE_ID)
REFERENCES EQUIPMENT_MAKE(ID)
ON DELETE SET NULL
/
ALTER TABLE EQUIPMENT_OPERATING_SYSTEM ADD CONSTRAINT
EQP_OS_EQP_FK FOREIGN KEY (EQUIPMENT_ID)
REFERENCES EQUIPMENT(ID)
ON DELETE CASCADE
ADD CONSTRAINT
EQP_OS_OS_VER_FK FOREIGN KEY (OPERATING_SYSTEM_VERSION_ID)
REFERENCES OPERATING_SYSTEM_VERSION(ID)
ON DELETE SET NULL
/
ALTER TABLE OPERATING_SYSTEM_VERSION ADD CONSTRAINT
OS_VER_OS_FK FOREIGN KEY (OPERATING_SYSTEM_ID)
REFERENCES OPERATING_SYSTEM(ID)
ON DELETE CASCADE

```

```

/
ALTER TABLE EQUIPMENT_SOFTWARE ADD CONSTRAINT
EQP_SFT_EQP_FK FOREIGN KEY (EQUIPMENT_ID)
REFERENCES EQUIPMENT(ID)
ON DELETE CASCADE
ADD CONSTRAINT
EQP_SFT_SFT_VER_FK FOREIGN KEY (SOFTWARE_VERSION_ID)
REFERENCES SOFTWARE_VERSION(ID)
ON DELETE SET NULL
/
ALTER TABLE SOFTWARE_VERSION ADD CONSTRAINT
SFT_VER_SFT_FK FOREIGN KEY (SOFTWARE_ID)
REFERENCES SOFTWARE(ID)
ON DELETE CASCADE
/
ALTER TABLE SOFTWARE_LICENSE ADD CONSTRAINT
SFT_LIC_SFT_VER_FK FOREIGN KEY (SOFTWARE_VERSION_ID)
REFERENCES SOFTWARE_VERSION(ID)
ON DELETE CASCADE
/
ALTER TABLE SOFTWARE_LICENSE_STATUS ADD CONSTRAINT
SFT_LIC_STA_SFT_LIC_FK FOREIGN KEY (SOFTWARE_LICENSE_ID)
REFERENCES SOFTWARE_LICENSE(ID)
ON DELETE CASCADE
/

COMMIT;

REM === INITIALISING SEQUENCES ===

PROMPT 'Creating sequences'

CREATE SEQUENCE CALLER_PHONE_ID_SEQ
/
CREATE SEQUENCE CALL_STATUS_ID_SEQ
/
CREATE SEQUENCE STAFF_ROLE_ID_SEQ
/
CREATE SEQUENCE STAFF_ID_SEQ
/
CREATE SEQUENCE CALL_ID_SEQ
/
CREATE SEQUENCE PROBLEM_CALL_ID_SEQ
/

```

```
CREATE SEQUENCE PROBLEM_ID_SEQ
/
CREATE SEQUENCE PROBLEM_SOLUTION_ID_SEQ
/
CREATE SEQUENCE PROBLEM_STATUS_ID_SEQ
/
CREATE SEQUENCE PROBLEM_ASSIGNMENT_ID_SEQ
/
CREATE SEQUENCE PROBLEM_TYPE_ID_SEQ
/
CREATE SEQUENCE PROBLEM_TYPE_SPECIALIST_ID_SEQ
/
CREATE SEQUENCE EQUIPMENT_ID_SEQ
/
CREATE SEQUENCE EQUIPMENT_MODEL_ID_SEQ
/
CREATE SEQUENCE EQUIPMENT_MAKE_ID_SEQ
/
CREATE SEQUENCE EQUIPMENT_TYPE_ID_SEQ
/
CREATE SEQUENCE EQUIPMENT_OPERATING_SYSTEM_ID_SEQ
/
CREATE SEQUENCE OPERATING_SYSTEM_VERSION_ID_SEQ
/
CREATE SEQUENCE OPERATING_SYSTEM_ID_SEQ
/
CREATE SEQUENCE EQUIPMENT_SOFTWARE_ID_SEQ
/
CREATE SEQUENCE SOFTWARE_VERSION_ID_SEQ
/
CREATE SEQUENCE SOFTWARE_ID_SEQ
/
CREATE SEQUENCE SOFTWARE_LICENSE_ID_SEQ
/
CREATE SEQUENCE SOFTWARE_LICENSE_STATUS_ID_SEQ
/

COMMIT;

REM === INITIALISING DATA ===

INSERT INTO CALL_STATUS VALUES (CALL_STATUS_ID_SEQ.NEXTVAL, 'MISSED');
INSERT INTO CALL_STATUS VALUES (CALL_STATUS_ID_SEQ.NEXTVAL, 'IN
PROGRESS');
```



```
INSERT INTO CALL_STATUS VALUES (CALL_STATUS_ID_SEQ.NEXTVAL, 'FINISHED');
```

```
INSERT INTO STAFF_ROLE VALUES (STAFF_ROLE_ID_SEQ.NEXTVAL, 'OPERATOR');
```

```
INSERT INTO STAFF_ROLE VALUES (STAFF_ROLE_ID_SEQ.NEXTVAL, 'SPECIALIST');
```

```
INSERT INTO PROBLEM_STATUS VALUES (  
  PROBLEM_STATUS_ID_SEQ.NEXTVAL, 'UNASSIGNED'  
);
```

```
INSERT INTO PROBLEM_STATUS VALUES (  
  PROBLEM_STATUS_ID_SEQ.NEXTVAL, 'ASSIGNED'  
);
```

```
INSERT INTO PROBLEM_STATUS VALUES (  
  PROBLEM_STATUS_ID_SEQ.NEXTVAL, 'IN PROGRESS'  
);
```

```
INSERT INTO PROBLEM_STATUS VALUES (  
  PROBLEM_STATUS_ID_SEQ.NEXTVAL, 'BLOCKED'  
);
```

```
INSERT INTO PROBLEM_STATUS VALUES (  
  PROBLEM_STATUS_ID_SEQ.NEXTVAL, 'RESOLVED'  
);
```

```
INSERT INTO PROBLEM_STATUS VALUES (  
  PROBLEM_STATUS_ID_SEQ.NEXTVAL, 'UNRESOLVED'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'CHARGER MALFUNCTION'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'BROKE SCREEN'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'RESTART LOOPING'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'HARDWARE MALFUNCTION'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'SOFTWARE MALFUNCTION'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'SOFTWARE LICENSE MALFUNCTION'  
);
```

```
INSERT INTO PROBLEM_TYPE VALUES (  
  PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'OTHER'
```

```
);  
INSERT INTO PROBLEM_TYPE VALUES (  
    PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'STUCK PRINTER'  
);  
INSERT INTO PROBLEM_TYPE VALUES (  
    PROBLEM_TYPE_ID_SEQ.NEXTVAL, 'BOOT MALFUNCTION'  
);
```

```
INSERT INTO EQUIPMENT_MAKE VALUES (  
    EQUIPMENT_MAKE_ID_SEQ.NEXTVAL, 'HP'  
);  
INSERT INTO EQUIPMENT_MAKE VALUES (  
    EQUIPMENT_MAKE_ID_SEQ.NEXTVAL, 'MICROSOFT'  
);  
INSERT INTO EQUIPMENT_MAKE VALUES (  
    EQUIPMENT_MAKE_ID_SEQ.NEXTVAL, 'ASUS'  
);  
INSERT INTO EQUIPMENT_MAKE VALUES (  
    EQUIPMENT_MAKE_ID_SEQ.NEXTVAL, 'APPLE'  
);  
INSERT INTO EQUIPMENT_MAKE VALUES (  
    EQUIPMENT_MAKE_ID_SEQ.NEXTVAL, 'LENOVO'  
);  
INSERT INTO EQUIPMENT_MAKE VALUES (  
    EQUIPMENT_MAKE_ID_SEQ.NEXTVAL, 'DELL'  
);
```

```
INSERT INTO EQUIPMENT_TYPE VALUES (  
    EQUIPMENT_TYPE_ID_SEQ.NEXTVAL, 'PRINTER'  
);  
INSERT INTO EQUIPMENT_TYPE VALUES (  
    EQUIPMENT_TYPE_ID_SEQ.NEXTVAL, 'LAPTOP'  
);  
INSERT INTO EQUIPMENT_TYPE VALUES (  
    EQUIPMENT_TYPE_ID_SEQ.NEXTVAL, 'DESKTOP'  
);  
INSERT INTO EQUIPMENT_TYPE VALUES (  
    EQUIPMENT_TYPE_ID_SEQ.NEXTVAL, 'PHONE'  
);
```

```
INSERT INTO OPERATING_SYSTEM VALUES (  
    OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 'WINDOWS'  
);  
INSERT INTO OPERATING_SYSTEM VALUES (  
    OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 'MAC OS'  
);
```

```
OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 'IOS'
);
INSERT INTO OPERATING_SYSTEM VALUES (
  OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 'MACOS'
);
INSERT INTO OPERATING_SYSTEM VALUES (
  OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 'LINUX'
);
INSERT INTO OPERATING_SYSTEM VALUES (
  OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 'OTHER'
);
```

```
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'PHOTOSHOP'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'EXCEL'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'WORD'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'SAFARI'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'OTHER'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'PRINT-NOW'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, 'VISUAL STUDIO'
);
INSERT INTO SOFTWARE VALUES (
  SOFTWARE_ID_SEQ.NEXTVAL, '3D MODELER'
);
```

```
INSERT INTO CALLER VALUES (
  '1111111111', 'John', 'Lenon'
);
INSERT INTO CALLER VALUES (
  '2222222222', 'Mark', 'Johnes'
);
INSERT INTO CALLER VALUES (
```

```

'3333333333', 'Cristiano', 'Ronaldo'
);
INSERT INTO CALLER VALUES (
'4444444444', 'Richard', 'Richardson'
);
INSERT INTO CALLER VALUES (
'5555555555', 'Valmor', 'Nostradamus'
);
INSERT INTO CALLER VALUES (
'6666666666', 'Johanne', 'Simon'
);
INSERT INTO CALLER VALUES (
'7777777777', 'Alexie', 'Fox'
);

INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '202-209-4424', '1111111111'
);
INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '202-332-2204', '2222222222'
);
INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '103-340-9481', '3333333333'
);
INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '222-421-1224', '4444444444'
);
INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '222-521-1225', '5555555555'
);
INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '214-869-9041', '6666666666'
);
INSERT INTO CALLER_PHONE VALUES (
CALLER_PHONE_ID_SEQ.NEXTVAL, '123-581-1752', '7777777777'
);

INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Cobal', 'Brian', 1
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Giovan', 'Denis', 1
);
INSERT INTO STAFF VALUES (

```

```

STAFF_ID_SEQ.NEXTVAL, 'Holy', 'Moly', 1
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Jenniffer', 'Coast', 2
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Sindy', 'Jupter', 2
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Lauren', 'White', 2
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Giselle', 'Brothyn', 2
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Maria', 'Jorreyn', 2
);
INSERT INTO STAFF VALUES (
STAFF_ID_SEQ.NEXTVAL, 'Forham', 'Brotheryn', 2
);

```

```

INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '1111111111', 1, NULL,
TO_TIMESTAMP('03-AUG-20 11:20:30.45 AM'),
TO_TIMESTAMP('03-AUG-20 11:59:20.25 AM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '1111111111', 3, 1,
TO_TIMESTAMP('03-AUG-20 3:20:30.45 PM'),
TO_TIMESTAMP('03-AUG-20 4:59:20.25 PM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '2222222222', 3, 2,
TO_TIMESTAMP('04-SEP-20 3:20:30.45 PM'),
TO_TIMESTAMP('04-SEP-20 4:59:20.25 PM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '3333333333', 3, 3,
TO_TIMESTAMP('01-SEP-20 10:20:30.45 PM'),
TO_TIMESTAMP('01-SEP-20 11:59:20.25 PM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '4444444444', 3, 3,
TO_TIMESTAMP('10-MAR-20 8:20:30.45 AM'),

```

```

TO_TIMESTAMP('10-MAR-20 9:59:20.25 AM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '222222222', 1, NULL,
TO_TIMESTAMP('03-AUG-20 11:20:30.45 AM'),
TO_TIMESTAMP('03-AUG-20 11:59:20.25 AM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '222222222', 3, 2,
TO_TIMESTAMP('03-AUG-19 3:30.45 PM'),
TO_TIMESTAMP('03-AUG-19 4:59:20.25 PM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '333333333', 3, 3,
TO_TIMESTAMP('04-JAN-20 3:20:30.45 PM'),
TO_TIMESTAMP('04-JAN-20 4:59:20.25 PM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '333333333', 3, 1,
TO_TIMESTAMP('20-JUN-20 4:20:30.45 PM'),
TO_TIMESTAMP('20-JUN-20 6:59:20.25 PM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '444444444', 3, 1,
TO_TIMESTAMP('10-MAR-20 4:20:30.45 AM'),
TO_TIMESTAMP('10-MAR-20 9:59:20.25 AM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '666666666', 3, 9,
TO_TIMESTAMP('03-FEB-20 10:20:30.45 AM'),
TO_TIMESTAMP('03-FEB-20 11:59:20.25 AM')
);
INSERT INTO CALL VALUES (
CALL_ID_SEQ.NEXTVAL, '777777777', 3, 8,
TO_TIMESTAMP('19-OCT-20 4:24:30.45 PM'),
TO_TIMESTAMP('19-OCT-20 4:59:20.25 PM')
);

INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'SUPER PRINTER',
TO_DATE('2000/02/04', 'yyyy/mm/dd'), 1, 1
);
INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'SURFACE 2018',

```

```

TO_DATE('2018/12/15', 'yyyy/mm/dd'), 2, 2
);
INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'ASUS DESKTOP 2019',
TO_DATE('2019/05/25', 'yyyy/mm/dd'), 3, 3
);
INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'IPHONE 2017',
TO_DATE('2017/03/05', 'yyyy/mm/dd'), 4, 4
);
INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'THINKPAD 2014',
TO_DATE('2014/08/09', 'yyyy/mm/dd'), 2, 5
);
INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'DELL INSPIRON 2017',
TO_DATE('2017/04/09', 'yyyy/mm/dd'), 2, 6
);
INSERT INTO EQUIPMENT_MODEL VALUES (
EQUIPMENT_MODEL_ID_SEQ.NEXTVAL, 'ASUS ZEN 2018',
TO_DATE('2018/03/11', 'yyyy/mm/dd'), 2, 3
);

/* printer */
INSERT INTO EQUIPMENT VALUES (
EQUIPMENT_ID_SEQ.NEXTVAL, '1234', 1
);
/* surface */
INSERT INTO EQUIPMENT VALUES (
EQUIPMENT_ID_SEQ.NEXTVAL, '4321', 2
);
/* asus desktop */
INSERT INTO EQUIPMENT VALUES (
EQUIPMENT_ID_SEQ.NEXTVAL, '12345', 3
);
/* iphone */
INSERT INTO EQUIPMENT VALUES (
EQUIPMENT_ID_SEQ.NEXTVAL, '54321', 4
);
/* Equipment without software */
INSERT INTO EQUIPMENT VALUES (
EQUIPMENT_ID_SEQ.NEXTVAL, '123414', 5
);
/* dell inspiron */

```

```

INSERT INTO EQUIPMENT VALUES (
    EQUIPMENT_ID_SEQ.NEXTVAL, '14y5125h1', 6
);
/* asus zen */
INSERT INTO EQUIPMENT VALUES (
    EQUIPMENT_ID_SEQ.NEXTVAL, '12491', 7
);

/* unassigned charger malfunction problem */
INSERT INTO PROBLEM VALUES (
    PROBLEM_ID_SEQ.NEXTVAL, 1, 1, 2,
    'The laptop battery is not charging the laptop anymore.',
    TO_TIMESTAMP('03-OCT-20 4:59:20.25 PM')
);
/* assigned broke screen problem */
INSERT INTO PROBLEM VALUES (
    PROBLEM_ID_SEQ.NEXTVAL, 2, 2, 3,
    'Some pixels on the screen are blue',
    TO_TIMESTAMP('04-SEP-20 4:59:20.25 PM')
);
/* restart looping problem in progress */
INSERT INTO PROBLEM VALUES (
    PROBLEM_ID_SEQ.NEXTVAL, 3, 3, 4,
    'The phone keeps restarting all the time',
    TO_TIMESTAMP('01-SEP-20 11:59:20.25 PM')
);
/* hardware malfunction problem resolved */
INSERT INTO PROBLEM VALUES (
    PROBLEM_ID_SEQ.NEXTVAL, 5, 4, 3,
    'The laptop memory chip has exploded',
    TO_TIMESTAMP('10-MAR-20 9:59:20.25 AM')
);
/* software malfunction problem resolved */
INSERT INTO PROBLEM VALUES (
    PROBLEM_ID_SEQ.NEXTVAL, 5, 5, 2,
    'Photoshop is crashing',
    TO_TIMESTAMP('03-AUG-19 4:59:20.25 PM')
);
/* software license malfunction problem resolved */
INSERT INTO PROBLEM VALUES (
    PROBLEM_ID_SEQ.NEXTVAL, 5, 6, 2,
    'Photoshop license has expired',
    TO_TIMESTAMP('04-JAN-20 4:59:20.25 PM')
);

```



```

);
/* unassigned boot malfunction */
INSERT INTO PROBLEM_VALUES (
  PROBLEM_ID_SEQ.NEXTVAL, 1, 9, 6,
  'Laptop is not booting at all',
  TO_TIMESTAMP('03-OCT-20 5:59:20.25 PM')
);
/* unassigned hardware malfunction */
INSERT INTO PROBLEM_VALUES (
  PROBLEM_ID_SEQ.NEXTVAL, 1, 4, 7,
  'Laptop is not booting at all',
  TO_TIMESTAMP('03-OCT-20 6:59:20.25 PM')
);

INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 2, 1
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 3, 2
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 4, 3
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 5, 4
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 7, 5
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 8, 6
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 11, 7
);
INSERT INTO PROBLEM_CALL_VALUES (
  PROBLEM_CALL_ID_SEQ.NEXTVAL, 12, 8
);

INSERT INTO PROBLEM_SOLUTION_VALUES (
  PROBLEM_SOLUTION_ID_SEQ.NEXTVAL, 4,
  'The laptop memory chip was substituted.',
  TO_TIMESTAMP('15-MAR-20 9:59:20.25 AM')
);

```

```
INSERT INTO PROBLEM_SOLUTION VALUES (  
  PROBLEM_SOLUTION_ID_SEQ.NEXTVAL, 5,  
  'A version upgrade fixed the problem.',  
  TO_TIMESTAMP('03-AUG-19 5:59:20.25 PM')  
);  
INSERT INTO PROBLEM_SOLUTION VALUES (  
  PROBLEM_SOLUTION_ID_SEQ.NEXTVAL, 6,  
  'A new license was added to the software.',  
  TO_TIMESTAMP('04-JAN-20 5:19:20.25 PM')  
);
```

```
INSERT INTO PROBLEM_ASSIGNMENT VALUES (  
  PROBLEM_ASSIGNMENT_ID_SEQ.NEXTVAL, 2, 4,  
  TO_TIMESTAMP('04-SEP-20 5:09:20.25 PM')  
);  
INSERT INTO PROBLEM_ASSIGNMENT VALUES (  
  PROBLEM_ASSIGNMENT_ID_SEQ.NEXTVAL, 3, 5,  
  TO_TIMESTAMP('01-SEP-20 12:59:20.25 PM')  
);  
INSERT INTO PROBLEM_ASSIGNMENT VALUES (  
  PROBLEM_ASSIGNMENT_ID_SEQ.NEXTVAL, 4, 6,  
  TO_TIMESTAMP('10-MAR-20 10:59:20.25 AM')  
);  
INSERT INTO PROBLEM_ASSIGNMENT VALUES (  
  PROBLEM_ASSIGNMENT_ID_SEQ.NEXTVAL, 5, 1,  
  TO_TIMESTAMP('03-AUG-19 5:59:20.25 PM')  
);  
INSERT INTO PROBLEM_ASSIGNMENT VALUES (  
  PROBLEM_ASSIGNMENT_ID_SEQ.NEXTVAL, 6, 2,  
  TO_TIMESTAMP('04-JAN-20 6:59:20.25 PM')  
);
```

```
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
  PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 7, 1  
);  
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
  PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 7, 2  
);  
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
  PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 7, 3  
);  
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
  PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 5, 4  
);
```

```
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
    PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 4, 5  
);  
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
    PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 6, 6  
);  
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
    PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 8, 8  
);  
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (  
    PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL, 9, 9  
);
```

```
INSERT INTO OPERATING_SYSTEM_VERSION VALUES (  
    OPERATING_SYSTEM_VERSION_ID_SEQ.NEXTVAL, '10',  
    TO_DATE('2017/05/05', 'yyyy/mm/dd'), 1  
);  
INSERT INTO OPERATING_SYSTEM_VERSION VALUES (  
    OPERATING_SYSTEM_VERSION_ID_SEQ.NEXTVAL, '9',  
    TO_DATE('2016/05/05', 'yyyy/mm/dd'), 2  
);  
INSERT INTO OPERATING_SYSTEM_VERSION VALUES (  
    OPERATING_SYSTEM_VERSION_ID_SEQ.NEXTVAL, 'MOJAVE',  
    TO_DATE('2018/08/05', 'yyyy/mm/dd'), 3  
);  
INSERT INTO OPERATING_SYSTEM_VERSION VALUES (  
    OPERATING_SYSTEM_VERSION_ID_SEQ.NEXTVAL, 'ARCHLINUX',  
    TO_DATE('2001/08/05', 'yyyy/mm/dd'), 4  
);  
INSERT INTO OPERATING_SYSTEM_VERSION VALUES (  
    OPERATING_SYSTEM_VERSION_ID_SEQ.NEXTVAL, 'HP PRINTER FIRMWARE',  
    TO_DATE('2001/08/05', 'yyyy/mm/dd'), 5  
);
```

```
INSERT INTO EQUIPMENT_OPERATING_SYSTEM VALUES (  
    EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 1, 5  
);  
INSERT INTO EQUIPMENT_OPERATING_SYSTEM VALUES (  
    EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 2, 1  
);  
INSERT INTO EQUIPMENT_OPERATING_SYSTEM VALUES (  
    EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 3, 4  
);  
INSERT INTO EQUIPMENT_OPERATING_SYSTEM VALUES (  
    EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 4, 1  
);
```

```

EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 4, 2
);
INSERT INTO EQUIPMENT_OPERATING_SYSTEM VALUES (
EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 6, 1
);
INSERT INTO EQUIPMENT_OPERATING_SYSTEM VALUES (
EQUIPMENT_OPERATING_SYSTEM_ID_SEQ.NEXTVAL, 7, 1
);

INSERT INTO SOFTWARE_VERSION VALUES (
SOFTWARE_VERSION_ID_SEQ.NEXTVAL, 'PS SUITE',
TO_DATE('2018/08/05', 'yyyy/mm/dd'), 1
);
INSERT INTO SOFTWARE_VERSION VALUES (
SOFTWARE_VERSION_ID_SEQ.NEXTVAL, 'PRO',
TO_DATE('2018/09/09', 'yyyy/mm/dd'), 2
);
INSERT INTO SOFTWARE_VERSION VALUES (
SOFTWARE_VERSION_ID_SEQ.NEXTVAL, 'BUSINESS',
TO_DATE('2017/12/05', 'yyyy/mm/dd'), 3
);
INSERT INTO SOFTWARE_VERSION VALUES (
SOFTWARE_VERSION_ID_SEQ.NEXTVAL, '1024',
TO_DATE('2015/12/12', 'yyyy/mm/dd'), 4
);
INSERT INTO SOFTWARE_VERSION VALUES (
SOFTWARE_VERSION_ID_SEQ.NEXTVAL, 'PRINTER STUDIO 2x',
TO_DATE('2015/04/05', 'yyyy/mm/dd'), 6
);

INSERT INTO EQUIPMENT_SOFTWARE VALUES (
EQUIPMENT_SOFTWARE_ID_SEQ.NEXTVAL, 1, 5
);
INSERT INTO EQUIPMENT_SOFTWARE VALUES (
EQUIPMENT_SOFTWARE_ID_SEQ.NEXTVAL, 2, 1
);
INSERT INTO EQUIPMENT_SOFTWARE VALUES (
EQUIPMENT_SOFTWARE_ID_SEQ.NEXTVAL, 2, 2
);
INSERT INTO EQUIPMENT_SOFTWARE VALUES (
EQUIPMENT_SOFTWARE_ID_SEQ.NEXTVAL, 3, 2
);
INSERT INTO EQUIPMENT_SOFTWARE VALUES (
EQUIPMENT_SOFTWARE_ID_SEQ.NEXTVAL, 4, 4
);

```

```

);

INSERT INTO SOFTWARE_LICENSE VALUES (
  SOFTWARE_LICENSE_ID_SEQ.NEXTVAL, '1249h51h5', 1
);
INSERT INTO SOFTWARE_LICENSE VALUES (
  SOFTWARE_LICENSE_ID_SEQ.NEXTVAL, '12094914h114', 2
);
INSERT INTO SOFTWARE_LICENSE VALUES (
  SOFTWARE_LICENSE_ID_SEQ.NEXTVAL, 'xias412124k', 3
);
INSERT INTO SOFTWARE_LICENSE VALUES (
  SOFTWARE_LICENSE_ID_SEQ.NEXTVAL, 'HE14i1i51i151', 4
);

INSERT INTO SOFTWARE_LICENSE_STATUS VALUES (
  SOFTWARE_LICENSE_STATUS_ID_SEQ.NEXTVAL, 'Y', 1
);
INSERT INTO SOFTWARE_LICENSE_STATUS VALUES (
  SOFTWARE_LICENSE_STATUS_ID_SEQ.NEXTVAL, 'Y', 2
);
INSERT INTO SOFTWARE_LICENSE_STATUS VALUES (
  SOFTWARE_LICENSE_STATUS_ID_SEQ.NEXTVAL, 'N', 3
);
INSERT INTO SOFTWARE_LICENSE_STATUS VALUES (
  SOFTWARE_LICENSE_STATUS_ID_SEQ.NEXTVAL, 'Y', 4
);

COMMIT;

```

Database test

Again, a **.sql** file can be downloaded on the link below:

[Help_desk_queries.sql](#)

```

/* 1) List the details of equipment and associated software */
SELECT e.id as EQUIPMENT_ID,
       em.NAME as EQUIPMENT_MODEL_NAME,
       em.RELEASE_DATE as EQUIPMENT_RELEASE_DATE,
       et.NAME as EQUIPMENT_TYPE,
       emake.NAME as EQUIPMENT_MAKE,
       s.name as SOFTWARE_NAME,

```

```

        sv.name as SOFTWARE_VERSION_NAME,
        sv.RELEASE_DATE as SOFTWARE_RELEASE_DATE
FROM equipment e
INNER JOIN equipment_model em ON (e.equipment_model_id = em.id)
INNER JOIN equipment_type et ON (em.equipment_type_id = et.id)
INNER JOIN equipment_make emake ON (em.equipment_make_id = emake.id)
LEFT OUTER JOIN equipment_software es ON (e.id = es.equipment_id)
LEFT OUTER JOIN software_version sv ON (sv.id = es.SOFTWARE_VERSION_ID)
LEFT OUTER JOIN software s ON (s.id = sv.SOFTWARE_ID);

```

SQL | All Rows Fetched: 8 in 1.636 seconds

EQUIPMENT_ID	EQUIPMENT_MODEL_NAME	EQUIPMENT_RELEASE_DATE	EQUIPMENT_TYPE	EQUIPMENT_MAKE	SOFTWARE_NAME	SOFTWARE_VERSION_NAME	SOFTWARE_RELEASE_DATE
2	SURFACE 2018	15-DEC-18	LAPTOP	MICROSOFT	PHOTOSHOP	PS SUITE	05-AUG-18
2	SURFACE 2018	15-DEC-18	LAPTOP	MICROSOFT	EXCEL	PRO	09-SEP-18
3	ASUS DESKTOP 2019	25-MAY-19	DESKTOP	ASUS	EXCEL	PRO	09-SEP-18
4	IPHONE 2017	05-MAR-17	PHONE	APPLE	SAFARI	1024	12-DEC-15
1	SUPER PRINTER	04-FEB-00	PRINTER	HP	PRINT-NOW	PRINTER STUDIO 2x	05-APR-15
6	DELL INSPIRON 2017	09-APR-17	LAPTOP	DELL	{null}	{null}	{null}
7	ASUS ZEN 2018	11-MAR-18	LAPTOP	ASUS	{null}	{null}	{null}
5	THINKPAD 2014	09-AUG-14	LAPTOP	LENOVO	{null}	{null}	{null}

/*

2) Produce a list of experts (support staff) for a given problem area.

First call "SELECT * FROM PROBLEM_TYPE" to know the available problem areas, then call the query below to find specialists for a given problem area.

*/

```

SELECT s.id as STAFF_ID,
       s.FIRST_NAME as FIRST_NAME,
       s.LAST_NAME as LAST_NAME,
       r.NAME as ROLE,
       pt.NAME as PROBLEM_TYPE_SPECIALISATION
FROM staff s
INNER JOIN staff_role r ON (r.id = s.staff_role_id)
INNER JOIN problem_type_specialist pts ON (pts.staff_id = s.id)
INNER JOIN problem_type pt ON (pts.problem_type_id = pt.id)
WHERE pt.NAME = '&PROBLEM_TYPE_NAME';

```

Enter value for PROBLEM_TYPE_NAME:




OK Cancel

SQL | All Rows Fetched: 1 in 0.303 seconds

	STAFF_ID	FIRST_NAME	LAST_NAME	ROLE	PROBLEM_TYPE_SPECIALISATION
1	5	Sindy	Juptyer	SPECIALIST	HARDWARE MALFUNCTION

Enter value for PROBLEM_TYPE_NAME:

OK Cancel




SQL | All Rows Fetched: 1 in 0.268 seconds

STAFF_ID	FIRST_NAME	LAST_NAME	ROLE	PROBLEM_TYPE_SPECIALISATION
1	4	Jennifer	Coast	SPECIALIST SOFTWARE MALFUNCTION

/*

3) Produce a list of all unresolved problems
with the assigned Specialist

*/

```
SELECT p.id as PROBLEM_ID,
       p.description as PROBLEM_DESCRIPTION,
       pt.name as PROBLEM_TYPE_NAME,
       s.first_name || ' ' || s.last_name as STAFF_FULL_NAME,
       ps.name AS PROBLEM_STATUS_NAME
FROM problem p
INNER JOIN problem_type pt ON (pt.id = p.problem_type_id)
INNER JOIN problem_status ps ON (ps.id = p.problem_status_id)
INNER JOIN problem_assignment pa ON (pa.problem_id = p.id)
LEFT OUTER JOIN staff s ON (pa.staff_id = s.id)
WHERE ps.name != 'RESOLVED';
```

SQL | All Rows Fetched: 2 in 1.512 seconds

PROBLEM_ID	PROBLEM_DESCRIPTION	PROBLEM_TYPE_NAME	STAFF_FULL_NAME	PROBLEM_STATUS_NAME
2	Some pixels on the screen are blue	BROKE SCREEN	Jennifer Coast	ASSIGNED
3	The phone keeps restarting all the time	RESTART LOOPING	Sindy Juptyer	IN PROGRESS

/*

4) Produce a list of problems reported by a member of staff,
the details of the support staff who attended the
problem and the solution provided by the support staff

Note the question is unclear as to whether problems without a solution
should be displayed. The query below excludes problems without a solution.

*/

```
SELECT c.id as CALL_ID,  
       clr.first_name || ' ' || clr.last_name as CALLER_NAME,  
       c_s.first_name || ' ' || c_s.last_name as CALL_OPERATOR_NAME,  
       p.description as PROBLEM_DESCRIPTION,  
       p_s.first_name || ' ' || p_s.last_name as PROBLEM_ASSIGNEE,  
       ps.description as PROBLEM_SOLUTION  
FROM call c  
INNER JOIN caller clr ON (clr.id = c.caller_id)  
INNER JOIN staff c_s ON (c_s.id = c.staff_id)  
INNER JOIN problem_call pc ON (pc.call_id = c.id)  
INNER JOIN problem p ON (p.id = pc.problem_id)  
INNER JOIN problem_assignment pa ON (pa.problem_id = p.id)  
INNER JOIN staff p_s ON (pa.staff_id = p_s.id)  
INNER JOIN problem_solution ps ON (ps.problem_id = p.id);
```

SQL All Rows Fetched: 3 in 1.509 seconds					
CALL_ID	CALLER_NAME	CALL_OPERATOR_NAME	PROBLEM_DESCRIPTION	PROBLEM_ASSIGNEE	PROBLEM_SOLUTION
7	Mark Johnes	Giovan Denis	Photoshop is crashing	Cobal Brian	A version upgrade fixed the problem.
5	Richard Richardson	Holy Moly	The laptop memory chip has exploded	Lauren White	The laptop memory chip was substituted.
8	Cristiano Ronaldo	Holy Moly	Photoshop license has expired	Giovan Denis	A new license was added to the software.

/*

5) Find (display) the average time taken to fix a fault for a
given problem area.

Note that I am counting the time the problem was created rather than
the time the call was picked up. Therefore this is more a measurement
of how the help desk department performs to solve a problem once this problem
is categorised rather than how much time operators spend on the line
clarifying a problem until the problem is solved. If we are interested
in the later, it is better to separate the concerns and do a query
measuring the length of calls for a certain problem type instead.

*/

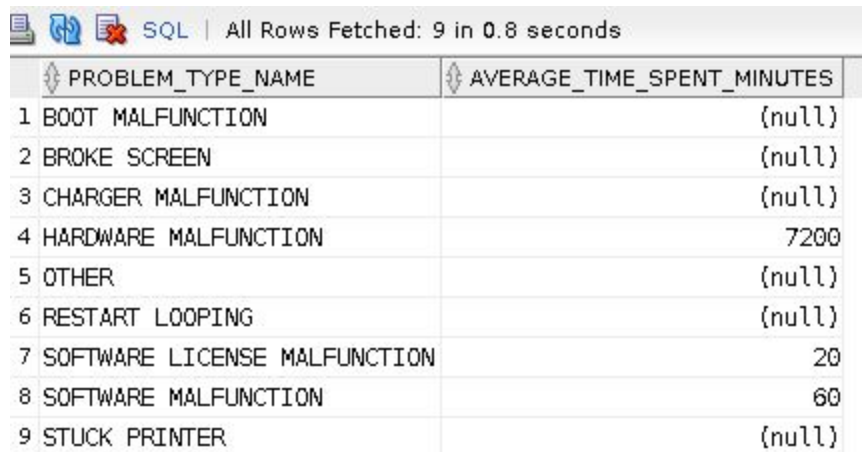
```
SELECT pt.NAME as PROBLEM_TYPE_NAME,
```



```

AVG(extract(day from (ps.datetime_created-p.datetime_created))*24*60
+ extract(hour from (ps.datetime_created-p.datetime_created))*60
+ extract(minute from (ps.datetime_created-p.datetime_created)))
as AVERAGE_TIME_SPENT_MINUTES
FROM problem_type pt
LEFT OUTER JOIN problem p ON (p.problem_type_id = pt.id)
LEFT OUTER JOIN problem_solution ps ON (ps.problem_id = p.id)
GROUP BY pt.NAME
ORDER BY pt.NAME;

```



PROBLEM_TYPE_NAME	AVERAGE_TIME_SPENT_MINUTES
1 BOOT MALFUNCTION	{null}
2 BROKE SCREEN	{null}
3 CHARGER MALFUNCTION	{null}
4 HARDWARE MALFUNCTION	7200
5 OTHER	{null}
6 RESTART LOOPING	{null}
7 SOFTWARE LICENSE MALFUNCTION	20
8 SOFTWARE MALFUNCTION	60
9 STUCK PRINTER	{null}

/*

6) Display a list of the most common problems and order them in according to the frequency of their occurrence.

*/

```

SELECT pt.NAME as PROBLEM_TYPE_NAME,
       COUNT(p.id) as OCCURRENCES
FROM problem_type pt
LEFT OUTER JOIN problem p ON (p.problem_type_id = pt.id)
GROUP BY pt.NAME
ORDER BY COUNT(p.id) DESC;

```

SQL All Rows Fetched: 9 in 0.293 seconds		
	PROBLEM_TYPE_NAME	OCCURRENCES
1	HARDWARE MALFUNCTION	2
2	CHARGER MALFUNCTION	1
3	BROKE SCREEN	1
4	SOFTWARE LICENSE MALFUNCTION	1
5	SOFTWARE MALFUNCTION	1
6	BOOT MALFUNCTION	1
7	RESTART LOOPING	1
8	OTHER	0
9	STUCK PRINTER	0

/*

7) Display the equipment with the most reported problems showing problem details.

Note: equipment details aren't asked, so only the equipment id is displayed on the resulting query.

*/

```
SELECT p.description, p.equipment_id
FROM problem p
WHERE p.equipment_id = (
    SELECT EQUIPMENT_ID FROM (
        SELECT p.equipment_id AS EQUIPMENT_ID, COUNT(p.equipment_id) as COUNTER
        FROM problem p
        GROUP BY p.equipment_id
        ORDER BY COUNTER DESC
        FETCH FIRST 1 ROWS ONLY
    )
);
```

SQL All Rows Fetched: 3 in 0.382 seconds		
	DESCRIPTION	EQUIPMENT_ID
	The laptop battery is not charging the laptop anymore.	2
	Photoshop is crashing	2
	Photoshop license has expired	2

/*

8) Produce a list of problems that have been solved by a helpdesk operator

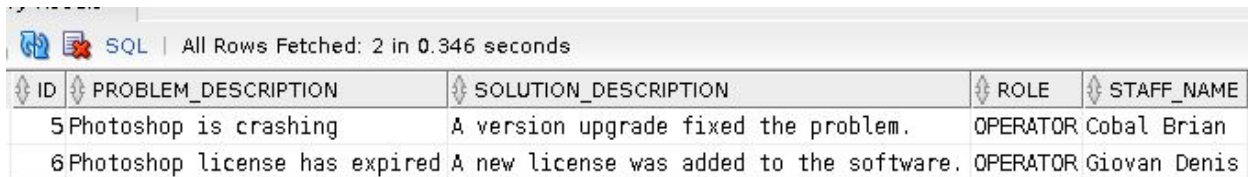
*/

```
SELECT p.id,
       p.description as PROBLEM_DESCRIPTION,
       ps.description as SOLUTION_DESCRIPTION,
```

```

        sr.name as ROLE,
        s.first_name || ' ' || s.last_name as STAFF_NAME
FROM problem p
INNER JOIN problem_assignment pa ON (pa.problem_id = p.id)
INNER JOIN staff s ON (pa.staff_id = s.id)
INNER JOIN problem_solution ps ON (ps.problem_id = p.id)
INNER JOIN staff_role sr ON (sr.id = s.staff_role_id)
WHERE sr.NAME = 'OPERATOR';

```



ID	PROBLEM_DESCRIPTION	SOLUTION_DESCRIPTION	ROLE	STAFF_NAME
5	Photoshop is crashing	A version upgrade fixed the problem.	OPERATOR	Cobal Brian
6	Photoshop license has expired	A new license was added to the software.	OPERATOR	Giovan Denis

```

/*
9) Add a new Specialist to the database
*/
INSERT INTO STAFF VALUES (
    STAFF_ID_SEQ.NEXTVAL, 'James', 'Smith',
    (SELECT sr.id FROM staff_role sr WHERE sr.NAME = 'SPECIALIST')
);
INSERT INTO PROBLEM_TYPE_SPECIALIST VALUES (
    PROBLEM_TYPE_SPECIALIST_ID_SEQ.NEXTVAL,
    (SELECT id FROM problem_type WHERE name = 'CHARGER MALFUNCTION'),
    (SELECT id FROM staff ORDER BY id DESC FETCH FIRST 1 ROWS ONLY)
);
COMMIT;

```

```
1 row inserted.
```

```
1 row inserted.
```

```
Commit complete.
```

```

/*
10) Assign a specialist to a given problem (it must be an unresolved,
unassigned and should find a matching Specialist)
*/

```

```

INSERT INTO PROBLEM_ASSIGNMENT VALUES (
    PROBLEM_ASSIGNMENT_ID_SEQ.NEXTVAL,
    (
        /* oldest unassigned problem */
        SELECT p.id
        FROM problem p

```

```

INNER JOIN problem_status ps ON (p.problem_status_id = ps.id)
LEFT OUTER JOIN problem_solution psol ON (psol.problem_id = p.id)
WHERE ps.name = 'UNASSIGNED' AND psol.id IS NULL
ORDER BY p.datetime_created FETCH FIRST 1 ROWS ONLY
),
(
/* least busy specialist in the area */
SELECT s.id
FROM staff s
INNER JOIN problem_type_specialist pts ON (pts.staff_id = s.id)
LEFT OUTER JOIN problem_assignment pa ON (pa.staff_id = s.id)
WHERE pts.problem_type_id = (
    SELECT p.problem_type_id
    FROM problem p
    INNER JOIN problem_status ps ON (p.problem_status_id = ps.id)
    LEFT OUTER JOIN problem_solution psol ON (psol.problem_id = p.id)
    WHERE (ps.name = 'UNASSIGNED' AND psol.id IS NULL)
    FETCH FIRST 1 ROWS ONLY
)
GROUP BY s.id
ORDER BY COUNT(pa.id) FETCH FIRST 1 ROWS ONLY
),
TO_CHAR(sysdate,'DD-MON-YYYY HH:MI:SS AM')
);
/* make sure the problem status is now ASSIGNED */
UPDATE problem
SET problem_status_id = (SELECT id FROM problem_status WHERE name = 'ASSIGNED')
WHERE problem.id = (
    /* oldest unassigned problem */
    SELECT p.id
    FROM problem p
    INNER JOIN problem_status ps ON (p.problem_status_id = ps.id)
    LEFT OUTER JOIN problem_solution psol ON (psol.problem_id = p.id)
    WHERE ps.name = 'UNASSIGNED' AND psol.id IS NULL
    ORDER BY p.datetime_created FETCH FIRST 1 ROWS ONLY
);
COMMIT;

```

```
1 row inserted.
```

```
1 row updated.
```

```
Commit complete.
```

```
/*
```

11) Update the Call-log with appropriate details when the problem has been resolved.

```
*/
```

```
INSERT INTO PROBLEM_SOLUTION VALUES (
```

```
    PROBLEM_SOLUTION_ID_SEQ.NEXTVAL,
```

```
/* oldest assigned and unresolve problem */
```

```
    (SELECT p.id
```

```
    FROM problem p
```

```
    INNER JOIN problem_status ps on (ps.id = p.problem_status_id)
```

```
    WHERE ps.name NOT IN ('UNASSIGNED', 'RESOLVED')
```

```
    ORDER BY p.datetime_created FETCH FIRST 1 ROWS ONLY
```

```
    ),
```

```
    'The battery was replaced and worked',
```

```
    TO_CHAR(sysdate,'DD-MON-YYYY HH:MI:SS AM')
```

```
);
```

```
UPDATE problem
```

```
SET problem_status_id = (SELECT id FROM problem_status WHERE name = 'RESOLVED')
```

```
WHERE problem.id = (
```

```
/* Latest problem solution */
```

```
SELECT ps.problem_id
```

```
FROM problem_solution ps
```

```
ORDER BY ps.datetime_created DESC FETCH FIRST 1 ROWS ONLY
```

```
);
```

```
COMMIT;
```

```
1 row inserted.
```

```
1 row updated.
```

```
Commit complete.
```

```
/*
```

12) An item of Equipment which, has previously had a fault reported, has now become redundant so Delete it from the database.

```
*/
```

```
DELETE FROM equipment
```

```
WHERE id = (  
  SELECT e.id FROM equipment e  
  INNER JOIN problem p ON (e.id = p.equipment_id)  
  ORDER BY id FETCH FIRST 1 ROWS ONLY  
);  
COMMIT;
```

```
| 1 row deleted.
```

```
| Commit complete.
```