

## REPORT ON DB OBJECT CREATION

<u>OBJECT NAME</u>	<u>NO. OF COLUMNS</u>
AUTHORS	2
BOOK	7
BOOKS_LOAD	7
BOOK_AUTHOR	3
BOOK_COPIES	3
BOOK_LOANS	6
BORROWERS	9
FINES	3
LIBRARY_BRANCH	3

### TO CREATE TABLE:

Go to SQL Workshop ->click on dropdown -> utilites data workshop

In data load click on text data

Click on new or existing table and select copy paste option in load from

Paste the data from the excel/notepad in the load data part and use ',' comma as separator

Enter table name and modify the column name or datatype or column length if required and click on next

Select a new column for primary key and generate a new sequence for it OR use an existing column for the primary key and select not generated option in primary key population

Click on load data button

Table will get created with loaded values and primary key

### FOR FOREIGN AND UNIQUE KEY GO TO CONSTRAINTS TAB OF THE PARTICULAR TABLE:

Click on create button

Select the constraint type as unique/foreign key and select cascade delete radio button option(this is to delete the constraints when dropping a table) and select the column to put constraint on

Select the table to reference it from and the column from that table to be referenced

Select next option and then click on finish to create the foreign key.

## 1. AUTHORS

```
CREATE TABLE "AUTHORS"
(
    "AUTHOR_ID" NUMBER,
    "AUTHOR_NAME" VARCHAR2(255),
    CONSTRAINT "AUTHORS_PK" PRIMARY KEY ("AUTHOR_ID")
    USING INDEX ENABLE
)
CREATE OR REPLACE EDITIONABLE TRIGGER "bi_AUTHORS"
before insert on "AUTHORS"
for each row
begin
    if :new."AUTHOR_ID" is null then
        select "AUTHORS_SEQ".nextval into :new."AUTHOR_ID" from sys.dual;
    end if;
end;

/
ALTER TRIGGER "bi_AUTHORS" ENABLE
/
```

## 2. BOOK

```
CREATE TABLE "BOOK"
(
    "ISBN" VARCHAR2(30),
    "ISBN13" NUMBER,
    "TITLE" VARCHAR2(255),
    "AUTHOR" VARCHAR2(255),
    "COVER" VARCHAR2(255),
    "PUBLISHER" VARCHAR2(255),
    "PAGES" NUMBER,
    CONSTRAINT "BOOK_PK" PRIMARY KEY ("ISBN")
    USING INDEX ENABLE
)
/

CREATE OR REPLACE EDITIONABLE TRIGGER "bi_BOOKS"
before insert on "BOOK"
for each row
begin
    if :new."ID" is null then
        select "BOOKS_SEQ".nextval into :new."ID" from sys.dual;
    end if;
end;

/
ALTER TRIGGER "bi_BOOKS" ENABLE/
```

## 3. BOOKS\_LOAD

```
CREATE TABLE "BOOKS_LOAD"
(
    "ISBN10" VARCHAR2(30),
    "ISBN13" NUMBER,
```

```

"TITLE" VARCHAR2(255),
"AUTHRO" VARCHAR2(255),
"COVER" VARCHAR2(255),
"PUBLISHER" VARCHAR2(255),
"PAGES" NUMBER
)

```

#### 4. BOOK\_AUTHORS

```

CREATE TABLE "BOOK_AUTHORS"
(
    "ID" NUMBER,
    "AUTHOR_ID" NUMBER,
    "ISBN" VARCHAR2(50),
    CONSTRAINT "BOOK_AUTHORS_PK" PRIMARY KEY ("ID")
    "BOOK_AUTHORS_FK1" foreign key ("ISBN") references "BOOK" ("ISBN")
    USING INDEX ENABLE
)
/
ALTER TABLE "BOOK_AUTHORS" ADD CONSTRAINT "BOOK_AUTHORS_FK1" FOREIGN KEY
("AUTHOR_ID")
    REFERENCES "AUTHORS" ("AUTHOR_ID") ON DELETE SET NULL ENABLE
/

CREATE OR REPLACE EDITIONABLE TRIGGER "bi_BOOK_AUTHORS"
before insert on "BOOK_AUTHORS"
for each row
begin
    if :new."ID" is null then
        select "BOOK_AUTHORS_SEQ1".nextval into :new."ID" from sys.dual;
    end if;
end;

/
ALTER TRIGGER "bi_BOOK_AUTHORS" DISABLE
/

```

#### 5. BOOK\_COPIES

```

CREATE TABLE "BOOK_COPIES"
(
    "BOOK_ID" VARCHAR2(30),
    "BRANCH_ID" NUMBER,
    "NO_OF_COPIES" NUMBER
)
alter table "BOOK_COPIES" add constraint
"BOOK_COPIES_CON" primary key ("BOOK_ID")
"BOOK_COPIES_CON" foreign key ("BOOK_ID") references "BOOK" ("ISBN")
"BOOK_COPIES" ADD CONSTRAINT "BOOK_COPIES_CON" FOREIGN KEY ("BRANCH_ID")
    REFERENCES "LIBRARY_BRANCH" ("BRANCH_ID") ENABLE
/

```

```

CREATE OR REPLACE EDITIONABLE TRIGGER "bi_BOOK_COPIES"
before insert on "BOOK_COPIES"
for each row
begin
if :new."ID" is null then
select "BOOK_COPIES_SEQ1".nextval into :new."ID" from sys.dual;
end if;
end;

/
ALTER TRIGGER "bi_BOOK_COPIES" DISABLE

/

```

## 6. BOOK\_LOAN

```

CREATE TABLE "BOOK_LOAN"
(
"BOOK_ID" VARCHAR2(30),
"CARD_NO" VARCHAR2(30),
"DATE_OUT" DATE,
"DUE_DATE" DATE,
"DATE_IN" DATE,
"LOAN_ID" NUMBER,
CONSTRAINT "BOOK_LOAN_PK" PRIMARY KEY ("LOAN_ID")
USING INDEX ENABLE
)
/
ALTER TABLE "BOOK_LOAN" ADD CONSTRAINT "BOOK_LOAN_CON" FOREIGN KEY
("BOOK_ID")
REFERENCES "BOOK" ("ISBN") ENABLE
/

CREATE OR REPLACE EDITIONABLE TRIGGER "PAIDCHECKCONDITION"
BEFORE INSERT
ON book_loan
FOR EACH ROW
DECLARE
paid_status_count number(10);
PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
select count(*) into paid_status_count from fines a join book_loan b on a.loan_id = b.loan_id
where card_no=:NEW.card_no and paid='Unpaid';
if paid_status_count <> 0
then
raise_application_error(-20000,'Fine Due');

end if;
Commit;
END;

/

```

```

ALTER TRIGGER "PAIDCHECKCONDITION" DISABLE
/

CREATE OR REPLACE EDITIONABLE TRIGGER "bi_BOOK_LOAN"
before insert on "BOOK_LOAN"
for each row
begin
if :new."LOAN_ID" is null then
select "AUTHOR_ID_SEQ".nextval into :new."LOAN_ID" from sys.dual;
end if;
end;

/
ALTER TRIGGER "bi_BOOK_LOAN" DISABLE
/
alter table "BOOK_LOAN" add constraint
"BOOK_LOAN_CON" foreign key ("CARD_NO") references "BORROWERS" ("ID0000ID")
/

```

## 7. BORROWERS

```

CREATE TABLE "BORROWERS"
(
    "ID0000ID" VARCHAR2(30),
    "SSN" VARCHAR2(30) NOT NULL ENABLE,
    "FIRST_NAME" VARCHAR2(30) NOT NULL ENABLE,
    "LAST_NAME" VARCHAR2(30) NOT NULL ENABLE,
    "EMAIL" VARCHAR2(255),
    "ADDRESS" VARCHAR2(30) NOT NULL ENABLE,
    "CITY" VARCHAR2(30) NOT NULL ENABLE,
    "STATE" VARCHAR2(30) NOT NULL ENABLE,
    "PHONE" VARCHAR2(30),
    CONSTRAINT "BORROWERS_PK" PRIMARY KEY ("ID0000ID")
USING INDEX ENABLE
)

CREATE OR REPLACE EDITIONABLE TRIGGER "BORROWERS_SSN_INSERT"
BEFORE INSERT
ON borrowers
FOR EACH ROW
DECLARE
    ssn_count number(10);
    pragma AUTONOMOUS_TRANSACTION;
BEGIN
select count(*)
into ssn_count
from borrowers
where ssn = :NEW.ssn;
if ssn_count > 0
then
    RAISE_APPLICATION_ERROR(-20000,'Duplicate SSN');
end if;
END;

/

```

```
ALTER TRIGGER "BORROWERS_SSN_INSERT" ENABLE
/
```

```
CREATE OR REPLACE EDITIONABLE TRIGGER "BORROWERS_T1"
BEFORE
insert or update or delete on "BORROWERS"
for each row
begin
create or replace TRIGGER borrowers_ssn_insert
BEFORE INSERT
  ON borrowers
  FOR EACH RO
DECLARE
  ssn_count number default 0;
  pragma AUTONOMOUS_TRANSACTION;
BEGIN
select count(*)
into ssn_count
from borrowers
where ssn = :NEW.ssn;
--dbms_output.put_line(:NEW.ssn);
if ssn_count > 0 then

  RAISE_APPLICATION_ERROR(-20000,'Duplicate SSN');

end if;

END;
end;
```

```
/
ALTER TRIGGER "BORROWERS_T1" DISABLE
/
```

```
CREATE OR REPLACE EDITIONABLE TRIGGER "bi_BORROWERS"
before insert on "BORROWERS"
for each row
begin
if :new."ID" is null then
  select "BORROWERS_SEQ".nextval into :new."ID" from sys.dual;
end if;
end;
```

```
/
ALTER TRIGGER "bi_BORROWERS" DISABLE
/
```

## 8. FINES

```
CREATE TABLE "FINES"
(
  "LOAN_ID" NUMBER(30,0),
  "FINE_AMT" FLOAT(30),
  "PAID" VARCHAR2(30),
```

```

        CONSTRAINT "FINES_CON" PRIMARY KEY ("LOAN_ID")
        USING INDEX ENABLE
    )
/
"FINES" add constraint
"FINES_FK" foreign key ("LOAN_ID") references "BOOK_LOAN"
("BOOK_ID","CARD_NO","DATE_IN","DATE_OUT","DUE_DATE","LOAN_ID")
/

```

## 9. LIBRARY BRANCH

```

CREATE TABLE "LIBRARY_BRANCH"
(
    "BRANCH_ID" NUMBER,
    "BRANCH_NAME" VARCHAR2(30),
    "ADDRESS" VARCHAR2(255),
    CONSTRAINT "LIBRARY_BRANCH_PK" PRIMARY KEY ("BRANCH_ID")
    USING INDEX ENABLE
)
CREATE OR REPLACE EDITIONABLE TRIGGER "bi_LIBRARY_BRANCH"
before insert on "LIBRARY_BRANCH"
for each row
begin
    if :new."ID" is null then
        select "LIBRARY_BRANCH_SEQ1".nextval into :new."ID" from sys.dual;
    end if;
end;

/
ALTER TRIGGER "bi_LIBRARY_BRANCH" ENABLE
/

```