

Технически университет - Варна
Дисциплина : Бази от данни



ДОКУМЕНТАЦИЯ

Изработил: Николай Иванов

КСТ гр. 1а курс 3

Фак. № 17621301

СЪДЪРЖАНИЕ

Задание.....	3
Релационен модел.....	4
Логически модел	5
Примерни данни	6
Код за генериране на таблиците.....	8
SQL команди	10
Тригери	13
Source code	15

Курсова работа БД

Задание

Да се проектира и реализира база от данни за **БИБЛИОТЕКА**, която да съхранява следната информация:

- Книга – наименование, автор, година на издаване, издателство, жанр;
- Служител – име, позиция, телефон
- Читател – име, телефон
- Заемане - книга, читател, служител, дата на заемане, срок

Правила:

- Всяка книга може да бъде зета текущо само от един читател,
- Всяка книга има един автор и едно издателство

Базата от данни трябва да е **нормализирана** и да позволява:

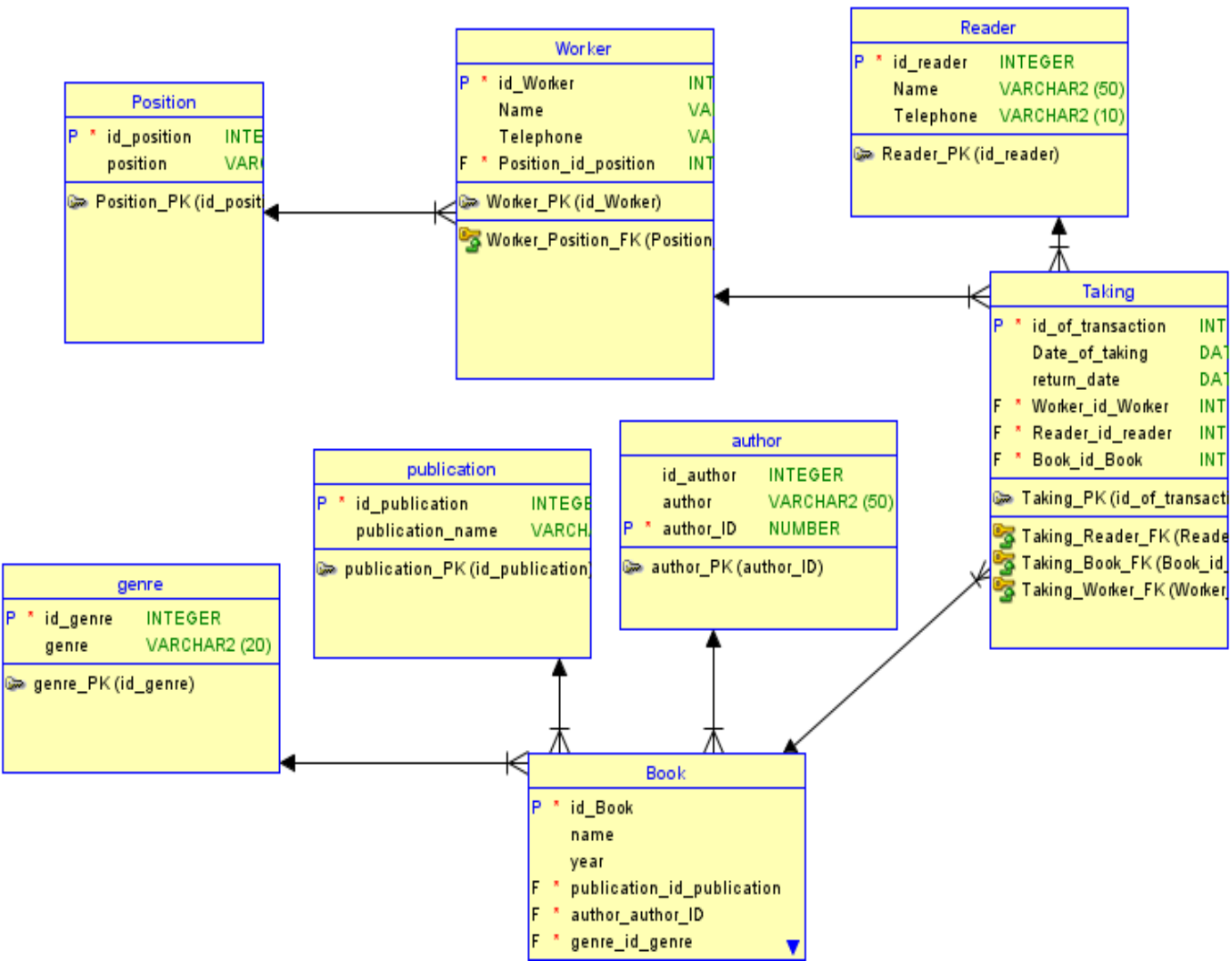
1. **Въвеждане и корекции** на данни.
2. **Търсене/заемане** на книга по: автор, ключови думи от наименованието, жанр.
3. Справки за:
 - Заети книги от читател.
 - Невърнати книги, подредени по дата.

Обяснителната записка към проекта трябва да съдържа:

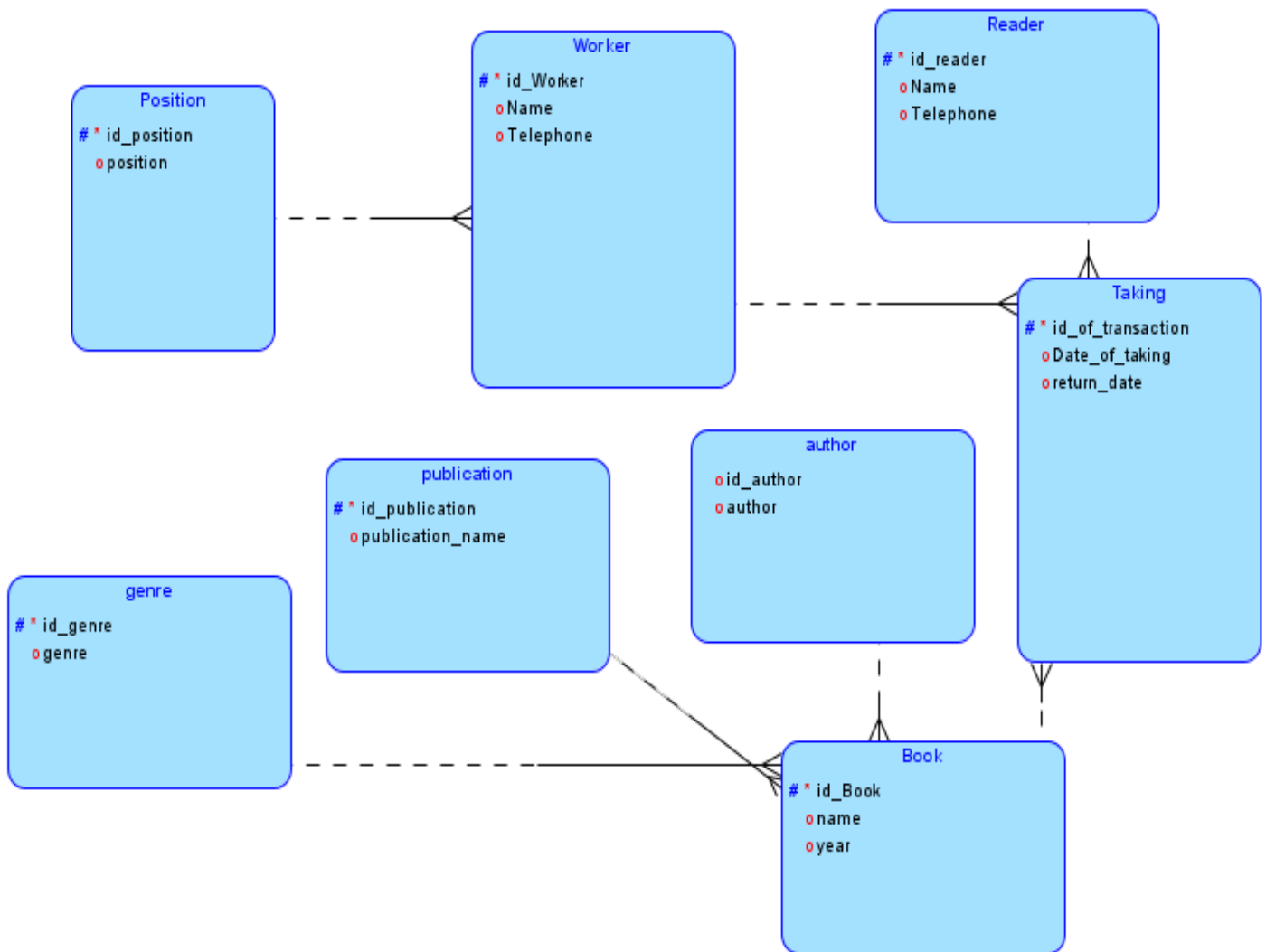
- Задание
- Моделите
- Примерни данни
- SQL – команди DDL, DML
- Резултати от изпълнението на заявките
- PL/SQL тригери

Модели на таблиците

Релационен модел:



Логически модел:



Примерни данни:

АВТОР:

ID_AUTHOR	AUTHOR
1	1 Issac Asimov
2	2 Sigmund Freud
3	3 George Orwell
4	4 Yordan Yovkov
5	5 J. R. Tolken
6	6 Sephen King
7	7 Pato
8	8 H.P Lovecraft
9	9 Friedrich Nietzsche
10	10 J. K. Rowling
11	11 Osamu Dazai
12	12 Fyodor Dostoyevsky

ЖАНЪР:

ID_GENRE	GENRE
1	1 Science fiction
2	2 Fantasy
3	3 Horror
4	4 Thriller
5	5 Romance
6	6 Hard boiled
7	7 Criminal
8	8 Hobby
9	9 Psychology
10	10 Philosophy
11	11 Drama

КНИГА:

ID_BOOK	NAME	YEAR	PUBLICATION_ID_PUBLICATION	AUTHOR_...	GENRE_ID...
1	1 Robots	21-OCT-60	7	1	1
2	2 Hobbit	04-NOV-19	4	5	2
3	3 Return of the king	01-NOV-30	5	5	2
4	4 Ratman	21-NOV-19	3	2	9
5	5 The Dunwich horror	21-DEC-01	4	8	3
6	8 No longer human	23-JAN-03	4	11	11
7	9 Idiot	16-AUG-00	4	12	11
8	10 Crime and Punishment	01-FEB-99	4	12	11
9	11 Harry Potter and the Goblet of Fire	08-JUL-00	7	10	2
10	12 Beyond Good and Evil	05-SEP-88	6	9	10
11	13 The Setting Sun	31-MAY-47	2	11	5
12	14 Dune	10-JUN-52	5	1	1
13	6 Dora	11-NOV-19	3	2	9
14	7 The republic	01-JAN-00	1	7	10

ИЗДАТЕЛСТВО:

	ID_PUBLICATION	PUBLICATION_NAME
1	1	Bulvest
2	2	Pravda
3	3	Anubis
4	4	Harper
5	5	West and East
6	6	Novo Slovo
7	7	Ravnica

СЛУЖИТЕЛСКИ ПОЗИЦИИ:

	ID_POSITION	POSITION
1	1	Checkout
2	2	Manager
3	3	Cleaner
4	4	IT support

ТРАНЗАНЦИЯ/ЗАЕМАНЕ:

	ID_OF_TRANSACTION	DATE_OF_TAKING	RETURN_DATE	WORKER_...	READER_I...	BOOK_ID_...
1	8	24-NOV-60	24-NOV-60	1	2	3
2	5	11-NOV-19	11-NOV-18	4	5	6
3	7	06-DEC-12	11-NOV-13	4	2	7
4	4	15-DEC-19	15-JAN-20	3	6	8
5	1	11-NOV-19	11-DEC-19	1	1	5
6	2	24-OCT-19	24-NOV-19	6	3	2
7	3	11-NOV-19	11-DEC-19	4	2	1

ЧИТАТЕЛ/КЛИЕНТ:

	ID_READER	NAME	TELEPHONE
1	1	Ivan Grigorov	0883459876
2	2	Teodora Petrova	0865652321
3	3	Svetoslav Mihailov	8799873338
4	4	Stanislav Jekov	0882333239
5	5	Ivan Vasilev	3278877844
6	6	Daniela Popova	9084284882
7	7	Natalia Raikova	7849987982
8	8	Mihaela Kuneva	2383984757
9	9	Petar Petrov	1832474777
10	10	Yoana Todorova	3578957755
11	11	Maria Ivanova	2983774500
12	12	Aleksandra Mircheva	0389280057
13	13	Ilia Nikolov	9201482099
14	14	Martin Petrov	8290039482

СЛУЖИТЕЛИ:

	ID_WORKER	NAME	TELEPHONE	POSITION_ID_POSITION
1	1	Nikolay Ivanov	0885452402	2
2	2	Georgi Petrov	0894341289	1
3	3	Yoana Ilieva	0885612310	1
4	4	Maria Vasiliva	0890019221	3
5	5	Hristo Minkov	8772321112	4
6	6	Nikola Aleksan...	0889891121	1
7	7	Todor Nikolov	0984225100	4
8	8	Tania Georgieva	0983221244	1

Код за генериране на таблиците:

-- Generated by Oracle SQL Developer Data Modeler 19.2.0.182.1216

-- at: 2019-10-21 08:41:33 PDT

-- site: Oracle Database 11g

-- type: Oracle Database 11g

```
CREATE TABLE author (  
  id_author INTEGER,  
  author VARCHAR2(50),  
  author_id NUMBER NOT NULL  
);
```

```
ALTER TABLE author ADD CONSTRAINT author_pk PRIMARY KEY ( author_id );
```

```
CREATE TABLE book (  
  id_book          INTEGER NOT NULL,  
  name             VARCHAR2(50),  
  year            DATE,  
  publication_id_publication INTEGER NOT NULL,  
  author_author_id NUMBER NOT NULL,  
  genre_id_genre   INTEGER NOT NULL  
);
```

```
ALTER TABLE book ADD CONSTRAINT book_pk PRIMARY KEY ( id_book );
```

```
CREATE TABLE genre (  
  id_genre INTEGER NOT NULL,  
  genre VARCHAR2(20)  
);
```

```
ALTER TABLE genre ADD CONSTRAINT genre_pk PRIMARY KEY ( id_genre );
```

```
CREATE TABLE position (  
  id_position INTEGER NOT NULL,  
  position VARCHAR2(30)  
);
```

```
ALTER TABLE position ADD CONSTRAINT position_pk PRIMARY KEY ( id_position );
```

```
CREATE TABLE publication (  
  id_publication INTEGER NOT NULL,  
  publication_name VARCHAR2(50)
```


);

ALTER TABLE publication ADD CONSTRAINT publication_pk PRIMARY KEY (id_publication);

```
CREATE TABLE reader (  
    id_reader INTEGER NOT NULL,  
    name      VARCHAR2(50),  
    telephone VARCHAR2(10)  
);
```

ALTER TABLE reader ADD CONSTRAINT reader_pk PRIMARY KEY (id_reader);

```
CREATE TABLE taking (  
    id_of_transaction INTEGER NOT NULL,  
    date_of_taking    DATE,  
    return_date       DATE,  
    worker_id_worker  INTEGER NOT NULL,  
    reader_id_reader  INTEGER NOT NULL,  
    book_id_book      INTEGER NOT NULL  
);
```

ALTER TABLE taking ADD CONSTRAINT taking_pk PRIMARY KEY (id_of_transaction);

```
CREATE TABLE worker (  
    id_worker      INTEGER NOT NULL,  
    name           VARCHAR2(50),  
    telephone      VARCHAR2(10),  
    position_id_position INTEGER NOT NULL  
);
```

ALTER TABLE worker ADD CONSTRAINT worker_pk PRIMARY KEY (id_worker);

```
ALTER TABLE book  
ADD CONSTRAINT book_author_fk FOREIGN KEY ( author_author_id )  
REFERENCES author ( author_id );
```

```
ALTER TABLE book  
ADD CONSTRAINT book_genre_fk FOREIGN KEY ( genre_id_genre )  
REFERENCES genre ( id_genre );
```

```
ALTER TABLE book  
ADD CONSTRAINT book_publication_fk FOREIGN KEY ( publication_id_publication )  
REFERENCES publication ( id_publication );
```

ALTER TABLE taking

**ADD CONSTRAINT taking_book_fk FOREIGN KEY (book_id_book)
REFERENCES book (id_book);**

ALTER TABLE taking

**ADD CONSTRAINT taking_reader_fk FOREIGN KEY (reader_id_reader)
REFERENCES reader (id_reader);**

ALTER TABLE taking

**ADD CONSTRAINT taking_worker_fk FOREIGN KEY (worker_id_worker)
REFERENCES worker (id_worker);**

ALTER TABLE worker

**ADD CONSTRAINT worker_position_fk FOREIGN KEY (position_id_position)
REFERENCES position (id_position);**

CREATE SEQUENCE author_author_id_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER author_author_id_trg BEFORE

INSERT ON author

FOR EACH ROW

WHEN (new.author_id IS NULL)

BEGIN

:new.author_id := author_author_id_seq.nextval;

END;

/

КОМАНДИ ЗА ВЪВЕЖДАНЕ НА ДАННИ В ТАБЛИЦИТЕ:

**INSERT INTO book VALUES(bookseq.nextval, '&book_name' , '&book_date' , &book_publication_id
,&book_author_id ,&book_genre_id);**

INSERT INTO reader VALUES(READERSEQ.nextval,'&reader_name','&reader_telephone');

**INSERT INTO worker VALUES (WORKERSEQ.nextval, '&worker_name' , '&worker_telephone' ,
&worker_position);**

INSERT INTO publication VALUES(PUBLICATIONSEQ.nextval, '&publication');

INSERT INTO position_ VALUES(POSITIONSEQ.nextval, '&position');

INSERT INTO genre VALUES(GENRESEQ.nextval, '&genre');

INSERT INTO author VALUES(AUTHORSEQ.nextval, '&author');

КОМАНДИ ЗА КОРЕГИРАНЕ НА ДАННИ ОТ ТАБЛИЦИТЕ:

UPDATE book SET name = '&name' WHERE id_book = &selected_id;

UPDATE reader SET name = '&name' WHERE id_reader = &selected_id;

UPDATE worker SET name = '&name' WHERE id_worker = &selected_id;

UPDATE publication SET publication_name = '&name' WHERE id_publication = &selected_id;

UPDATE author SET author = '&name' WHERE id_author = &selected_id;

UPDATE genre SET genre = '&genre' WHERE id_genre = &selected_id;

UPDATE position_set position = '&position' WHERE id_position = &selected_id;

КОМАНДИ ЗА :

Търсене на книга по ключова дума:

```
SELECT BOOK.name , BOOK.year ,publication.publication_name ,AUTHOR.AUTHOR
,genre.genre FROM BOOK
INNER JOIN GENRE
ON book.genre_id_genre = genre.id_genre
INNER JOIN AUTHOR
ON book.AUTHOR_AUTHOR_ID = AUTHOR.id_author
INNER JOIN PUBLICATION
ON book.PUBLICATION_ID_PUBLICATION = publication.id_publication -- vsichko
WHERE BOOK.NAME LIKE '%&selected%';
```

При въведен "republic" на SQL изкарва "The republic"

	NAME	YEAR	PUBLICATION_NAME	AUTHOR	GENRE
1	The republic	01-JAN-00	Bulvest	Pato	Philosophy

Търсене на книга по жанр:

```
SELECT BOOK.name , BOOK.year ,publication.publication_name ,AUTHOR.AUTHOR
,genre.genre FROM BOOK
INNER JOIN GENRE
ON book.genre_id_genre = genre.id_genre
INNER JOIN AUTHOR
ON book.AUTHOR_AUTHOR_ID = AUTHOR.id_author
INNER JOIN PUBLICATION
ON book.PUBLICATION_ID_PUBLICATION = publication.id_publication
WHERE genre_id_genre = &Genre_ID;
```

При въведено "1" (асоцийран с sci-fi жанра) SQL изкарва:

	NAME	YEAR	PUBLICATION_NAME	AUTHOR	GENRE
1	Dune	10-JUN-52	West and East	Issac Asimov	Science fiction
2	Robots	21-OCT-60	Ravnica	Issac Asimov	Science fiction

Търсене на книга по автор:

```
SELECT BOOK.name , BOOK.year ,publication.publication_name ,AUTHOR.AUTHOR
,genre.genre FROM BOOK
INNER JOIN GENRE
ON book.genre_id_genre = genre.id_genre
INNER JOIN AUTHOR
ON book.AUTHOR_AUTHOR_ID = AUTHOR.id_author
INNER JOIN PUBLICATION
ON book.PUBLICATION_ID_PUBLICATION = publication.id_publication
WHERE author_author_id = &Author_ID;
```

При въведено "8" (асоцийран с автора H.P Lovecraft) SQL изкарва:

	NAME	YEAR	PUBLICATION_NAME	AUTHOR	GENRE
1	The Dunwich horror	21-DEC-01	Harper	H.P Lovecraft	Horror

Книги взети от читател:

```
SELECT taking.book_id_book, book.name FROM taking
INNER JOIN book
ON taking.book_id_book = book.id_book
WHERE taking.reader_id_reader = &reader_ID;
```

При въведен "1" (асоцийран с първи читател) SQL изкарва:

	BOOK_ID_BOOK	NAME
1	5	The Dunwich horror

Невърнати книги, подредени по дата:

```
SELECT * FROM taking ORDER BY return_date;
```

	ID_OF_TRANSACTION	DATE_OF_TAKING	RETURN_DATE	WORKER_ID_WORKER	READER_ID_READER	BOOK_ID_BOOK
1	8	24-NOV-60	24-NOV-60	1	2	3
2	7	06-DEC-12	11-NOV-13	4	2	7
3	5	11-NOV-19	11-NOV-18	4	5	6
4	2	24-OCT-19	24-NOV-19	6	3	2
5	1	11-NOV-19	11-DEC-19	1	1	5
6	3	11-NOV-19	11-DEC-19	4	2	1
7	4	15-DEC-19	15-JAN-20	3	6	8

ТРИГЕРИ:

Тригер за генериране на ID за нов работник:

```
create or replace trigger WORKERTRIGGER
before insert on "HR"."WORKER"
for each row
begin
    if inserting then
        if :NEW."ID_WORKER" is null then
            select WORKERSEQ.nextval into :NEW."ID_WORKER" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за нов автор:

```
create or replace trigger AuthorTrigger
before insert on "HR"."AUTHOR"
for each row
begin
    if inserting then
        if :NEW."ID_AUTHOR" is null then
            select AUTHORSEQ.nextval into :NEW."ID_AUTHOR" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за нов жанър:

```
create or replace trigger genreTrigger
before insert on "HR"."GENRE"
for each row
begin
    if inserting then
        if :NEW."ID_GENRE" is null then
            select GENRESEQ.nextval into :NEW."ID_GENRE" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за нова позиция:

```
create or replace trigger POSITIONTRIGGER
before insert on "HR"."POSITION_"
for each row
begin
    if inserting then
        if :NEW."ID_POSITION" is null then
            select POSITIONSEQ.nextval into :NEW."ID_POSITION" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за нова публикация:

```
create or replace trigger PUBLICATIONSEQ
before insert on "HR"."PUBLICATION"
for each row
begin
    if inserting then
        if :NEW."ID_PUBLICATION" is null then
            select POSITIONSEQ.nextval into :NEW."ID_PUBLICATION" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за нов читател:

```
create or replace trigger READERTRIGGER
before insert on "HR"."READER"
for each row
begin
    if inserting then
        if :NEW."ID_READER" is null then
            select READERSEQ.nextval into :NEW."ID_READER" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за ново заемане:

```
create or replace trigger TAKINGTRIGGER
before insert on "HR"."TAKING"
for each row
begin
    if inserting then
        if :NEW."ID_OF_TRANSACTION" is null then
            select TAKINGSEQ.nextval into :NEW."ID_OF_TRANSACTION" from dual;
        end if;
    end if;
end;
```

Тригер за генериране на ID за нова книга:

```
create or replace trigger newnwnewtrigger
before insert on "HR"."BOOK"
for each row
begin
    if inserting then
        if :NEW."ID_BOOK" is null then
            select BOOKSEQ.nextval into :NEW."ID_BOOK" from dual;
        end if;
    end if;
end;
```

Source code:

```
ALTER TABLE taking ADD CONSTRAINT taking_bbq UNIQUE (book_id_book);
```

```
INSERT INTO book VALUES( bookseq.nextval, '&book_name' , '&book_date' ,&book_publication_id, &
book_author_id, &book_genre_id);
INSERT INTO reader VALUES(READERSEQ.nextval,'&reader_name','&reader_telephone');
INSERT INTO worker VALUES (WORKERSEQ.nextval, '&worker_name', '&worker_telephone' ,&worker_position);
INSERT INTO publication VALUES(PUBLICATIONSEQ.nextval,'&publication');
INSERT INTO position_ VALUES(POSITIONSEQ.nextval,'&position');
INSERT INTO genre VALUES(GENRESEQ.nextval,'&genre');
INSERT INTO author VALUES(AUTHORSEQ.nextval,'&author');
INSERT INTO taking VALUES( &transation_id, '&taking_date' , '&return_date' ,&worker_id,&reader_id,&book_idf);
```

```
UPDATE book SET name = '&name' WHERE id_book = &selected_id;
UPDATE reader SET name = '&name' WHERE id_reader = &selected_id;
UPDATE worker SET name = '&name' WHERE id_worker = &selected_id;
UPDATE publication SET publication_name = '&name' WHERE id_publication = &selected_id;
UPDATE author SET author = '&name' WHERE id_author = &selected_id;
UPDATE genre SET genre = '&genre' WHERE id_genre = &selected_id;
UPDATE position_ set position = '&position' WHERE id_position = &selected_id;
```

```
SELECT BOOK.name , BOOK.year ,publication.publication_name ,AUTHOR.AUTHOR ,genre.genre FROM BOOK
INNER JOIN GENRE
ON book.genre_id_genre = genre.id_genre
INNER JOIN AUTHOR
ON book.AUTHOR_AUTHOR_ID = AUTHOR.id_author
INNER JOIN PUBLICATION
ON book.PUBLICATION_ID_PUBLICATION = publication.id_publication
WHERE BOOK.NAME LIKE '%&selected%';
```

```
SELECT BOOK.name , BOOK.year ,publication.publication_name ,AUTHOR.AUTHOR ,genre.genre FROM BOOK
INNER JOIN GENRE
ON book.genre_id_genre = genre.id_genre
INNER JOIN AUTHOR
ON book.AUTHOR_AUTHOR_ID = AUTHOR.id_author
INNER JOIN PUBLICATION
ON book.PUBLICATION_ID_PUBLICATION = publication.id_publication
WHERE genre_id_genre = &Genre_ID;
```

```
SELECT BOOK.name , BOOK.year ,publication.publication_name ,AUTHOR.AUTHOR ,genre.genre FROM BOOK
INNER JOIN GENRE
ON book.genre_id_genre = genre.id_genre
INNER JOIN AUTHOR
ON book.AUTHOR_AUTHOR_ID = AUTHOR.id_author
INNER JOIN PUBLICATION
ON book.PUBLICATION_ID_PUBLICATION = publication.id_publication
WHERE author_author_id = &Author_ID;
```

```
SELECT taking.book_id_book, book.name FROM taking
INNER JOIN book
ON taking.book_id_book = book.id_book
WHERE taking.reader_id_reader = &reader_ID;

SELECT * FROM taking ORDER BY return_date;
```