

## Database of the rental book

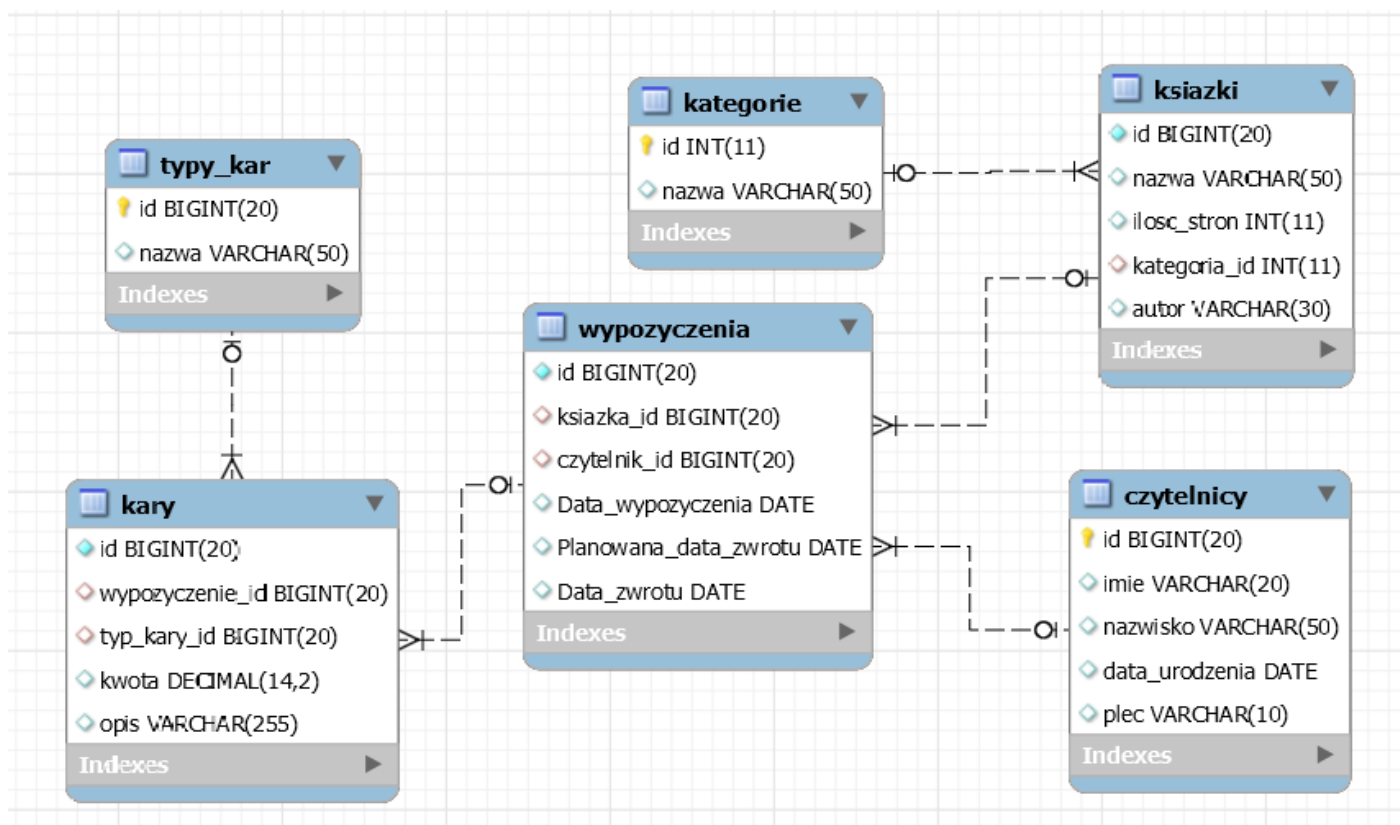


Fig. 1 Above screenshot show relations between tables in the MySQL Workbench.

	id	nazwa	ilosc_stron	kategoria_id	autor
▶	1	Cień	434	1	Przechrztła Adam
	2	Podejrzany	320	3	Świsł Paulina
	3	Outsider	640	2	King Stephen
	4	Tango	208	5	Mrozek Sławomir
	5	Miała dzikie serce	232	4	Atticus
	6	Dramaty	280	5	Masłowska Dorota
	7	Cztery żywioły. Tom 4. Czerwony Pająk	816	3	Bonda Katarzyna
	8	Mitologia nordycka	208	1	Gaiman Neil
	9	Policja	386	5	Mrozek Sławomir
	10	To	1150	2	King Stephen
	11	Wirus	320	2	Masterton Graham
	12	I że cię nie opuszczę	488	3	Richmond Michelle
	13	Otchłań. Księga 2	565	1	Brett Peter V.
	14	Mleko i miód. Milk and Honey	416	4	Rupi Kaur
	15	Zew Cthulhu	296	2	Lovecraft Howard Phillips

Fig. 2 Above screenshot show the table with books

	id	nazwa
▶	1	Fantastyka
	2	Horror
	3	Kryminal
	4	Poezja
	5	Dramat

Fig. 3. Screenshot from the book categories table

	id	imie	nazwisko	data_urodzenia	plec
▶	1	Malgorzata	Janiszewska	1985-04-02	Kobieta
	2	Kinga	Kurek	1947-03-15	Kobieta
	3	Kamil	Michalak	1988-07-16	Mezczyzna
	4	Joanna	Paluch	1980-05-22	Kobieta
	5	Zofia	Kurowska	1990-06-01	Kobieta
	6	Antoni	Lipinski	1990-04-15	Mezczyzna
	7	Kaja	Domagala	1990-09-19	Kobieta
	8	Szymon	Zak	1975-11-25	Mezczyzna
	9	Michal	Kasprzak	1989-10-13	Mezczyzna

Fig. 4. Screenshot from the readers table

	id	ksiazka_id	czytelnik_id	Data_wypozyczenia	Planowana_data_zwrotu	Data_zwrotu
▶	1	7	3	2017-11-18	2018-01-17	2018-01-12
	2	5	6	2018-06-05	2018-08-04	NULL
	3	3	4	2017-09-29	2017-11-28	2017-11-23
	4	11	8	2018-04-10	2018-06-09	2018-06-04
	5	15	1	2018-01-11	2018-03-12	NULL
	6	7	9	2018-05-01	2018-06-30	2018-06-25
	7	8	1	2018-03-05	2018-05-04	2018-04-29
	8	9	4	2018-02-23	2018-04-24	NULL
	9	8	6	2017-12-16	2018-02-14	NULL
	10	8	2	2017-08-22	2017-10-21	2017-10-16
	11	3	3	2017-08-22	2017-10-21	2017-10-25
	12	4	1	2018-02-05	2018-04-06	NULL
	13	6	6	2017-11-08	2018-01-07	NULL
	14	2	4	2018-02-28	2018-04-29	NULL
	15	13	4	2018-02-23	2018-04-24	NULL
	16	7	7	2018-04-10	2018-06-09	NULL
	17	4	4	2017-12-16	2018-02-14	2018-01-23
	18	8	5	2018-01-11	2018-03-12	2018-02-05
	19	11	6	2018-06-05	2018-08-04	NULL
	20	14	4	2017-04-21	2017-06-20	NULL

Fig. 5. Above Screenshot show table with rentals

	id	wypozyczenie_id	typ_kary_id	kwota	opis
▶	1	3	1	40.00	Nie nadaje sie do ponownego uzytku
	2	5	2	5.00	NULL
	3	7	2	5.00	NULL
	4	12	3	10.00	Zalanie kilku stron
	5	14	2	5.00	NULL
	6	20	2	5.00	NULL

Fig. 6. Above Screenshot show table with penalties

	id	nazwa
▶	1	Zniszczenie
	2	Brak terminowego zdania
	3	Znaczne pogorszenie stanu

Fig. 7. Above Screenshot show table with type of penalties

### 1.Zestawienie książek, które nie zostały nigdy wypożyczone

SELECT

nazwa

FROM

ksiazki

LEFT JOIN

wypozyczenia ON ksiazki.id = wypozyczenia.ksiazka\_id

WHERE

wypozyczenia.ksiazka\_id IS NULL;

### 2. Zestawienie książek, które zostały wypożyczone

SELECT

ksiazki.id, nazwa

FROM

ksiazki

INNER JOIN

wypozyczenia ON ksiazki.id = wypozyczenia.ksiazka\_id

GROUP BY nazwa

ORDER BY ksiazki.id;

### 3.Zestawienie czytelników, którzy posiadają aktualnie wypożyczoną książkę, nie dokonali jej zwrócenia.

SELECT

czytelnicy.id, imie, nazwisko

FROM

czytelnicy

LEFT JOIN

wypozyczenia ON czytelnicy.id = wypozyczenia.czytelnik\_id

WHERE

Data\_zwrotu IS NULL

GROUP BY czytelnicy.id

ORDER BY czytelnicy.id;

### 4.. Kategorie wraz z ilością książek przypisanych do danej kategorii.

SELECT

kategorie.nazwa, COUNT(\*) AS ilosc

FROM

kategorie

LEFT JOIN

ksiazki ON kategorie.id = ksiazki.kategoria\_id

GROUP BY kategorie.nazwa

ORDER BY kategorie.id;

#### 5. Pierwszych pięciu czytelników, którzy posiadają największą ilość wypożyczeń.

SELECT

czytelnicy.id,

czytelnicy.imie,

czytelnicy.nazwisko,

COUNT(\*) AS ilosc\_wypozycczen

FROM

czytelnicy

LEFT JOIN

wypozycczenia ON czytelnicy.id = wypozycczenia.czytelnik\_id

GROUP BY czytelnicy.id

ORDER BY ilosc\_wypozycczen DESC

LIMIT 5;

#### 6. Czytelnicy wraz z sumą nałożonych na nich kar. Interesują nas tylko Ci, którzy posiadają więcej niż 1 karę, Ci o najwyższej sumie kar mają być wyświetlani jako pierwsi.

SELECT

czytelnicy.id,

czytelnicy.imie,

czytelnicy.nazwisko,

SUM(kary.kwota) AS naleznosc

FROM

czytelnicy

LEFT JOIN

wypozycczenia ON czytelnicy.id = wypozycczenia.czytelnik\_id

INNER JOIN

kary ON kary.wypozycczenie\_id = wypozycczenia.id

GROUP BY czytelnicy.id

ORDER BY naleznosc DESC;

**7. Z jakiego typu kar posiadamy największe wpływy, wyświetl nazwę kary wraz z sumą kwot.**

SELECT

nazwa, SUM(kary.kwota) AS suma\_kar, COUNT(\*) AS ilosc\_kar

FROM

typy\_kar

LEFT JOIN

kary ON typy\_kar.id = kary.typ\_kary\_id

GROUP BY typy\_kar.id;

**8. Czytelnicy w jakim wieku posiadają największą ilość wypożyczeń? Zwróć wiek wraz z ilością wypożyczeń.**

SELECT

czytelnicy.id,

czytelnicy.imie,

czytelnicy.nazwisko,

czytelnicy.data\_urodzenia,

COUNT(\*) AS ilosc\_wyp\_ksiazek

FROM

czytelnicy

LEFT JOIN

wypozyczenia ON czytelnicy.id = wypozyczenia.czytelnik\_id

GROUP BY czytelnicy.id

ORDER BY ilosc\_wyp\_ksiazek DESC , data\_urodzenia;

**9. Która płeć wypożycza większą ilość książek oraz jakie ma to odniesienie do wysokości naliczonych kar. Zwróć płeć, ilość wypożyczeń oraz sumę naliczonych kar.**

SELECT

czytelnicy.plec,

COUNT(\*) AS ilosc\_wypozyczen,

SUM(kary.kwota) AS suma\_kar

FROM

czytelnicy

LEFT JOIN

wypozyczenia ON wypozyczenia.czytelnik\_id = czytelnicy.id

LEFT JOIN

kary ON kary.wypozyczenie\_id = wypozyczenia.id

GROUP BY plec;

**10. Wyświetl książki, którym zostało mniej niż tydzień do planowej daty ich zwrotu.**

**Założenie: dzisiejsza data to '2018-04-20'**

SELECT

```
ksiazki.nazwa,  
wypozyczenia.Planowana_data_zwrotu,  
DATEDIFF('2018-04-20',  
wypozyczenia.Planowana_data_zwrotu) AS dni_do_zwrotu
```

FROM

```
ksiazki  
LEFT JOIN  
wypozyczenia ON wypozyczenia.ksiazka_id = ksiazki.id
```

WHERE

```
DATEDIFF('2018-04-20',  
wypozyczenia.Planowana_data_zwrotu) < 7  
AND DATEDIFF('2018-04-20',  
wypozyczenia.Planowana_data_zwrotu) > - 7;
```

**11. Wyświetl książki, które nie zostały zwrócone w terminie oraz nie zostały dla nich jeszcze naliczone kary z tytułu owego przekroczenia.**

SELECT

```
ksiazki.id, ksiazki.nazwa, wypozyczenia.id
```

FROM

```
ksiazki  
INNER JOIN  
wypozyczenia ON wypozyczenia.ksiazka_id = ksiazki.id  
LEFT JOIN  
kary ON kary.wypozyczenie_id = wypozyczenia.id
```

WHERE

```
wypozyczenia.Data_zwrotu > wypozyczenia.Planowana_data_zwrotu  
OR wypozyczenia.Data_zwrotu IS NULL  
AND kary.id IS NULL;
```

## Procedures

-- Below procedure show all readers from 'czytelnicy' table;

drop procedure if exists readers;

create procedure readers()

select \* from czytelnicy;

call readers();

-- Below procedure show single reader from 'czytelnicy' table

drop procedure if exists reader\_selection;

create procedure reader\_selection(in nr\_id int)

select \* from czytelnicy where id = nr\_id;

call reader\_selection(5);

drop procedure if exists add\_reader;

-- Below procedure add reader into 'czytelnicy' table

Delimiter \\\

create procedure add\_reader (in Name\_ varchar(10),in Surname\_ varchar(20),in Date\_of\_birth Date, in Gender varchar(10))

begin

insert into czytelnicy(imie, nazwisko, data\_urodzenia, plec) values(Name\_, Surname\_, Date\_of\_birth, Gender);

call readers();

end \\\

delimiter ;

call add\_reader('Marcin','Bury','2000-07-25','Mezcyzna'); -- Calling above 'add\_reader' procedure

drop procedure if exists add\_readers;

-- Below procedure add many readers with to same personaldata into 'czytelnicy' table

Delimiter \\\

CREATE PROCEDURE add\_readers(in nr\_readers int,in Name\_ varchar(20),in Surname\_ varchar(20),in Date\_of\_birth Date, in Gender varchar(10))

BEGIN

WHILE nr\_readers > 0 DO

call add\_reader(Name\_, Surname\_, Date\_of\_birth, Gender);

```

    SET nr_readers = nr_readers - 1;

END WHILE;

call readers();

END \\\n
delimiter ;

call add_readers (3,'Marcin','Bury','2000-00-00','Mezczynna'); -- Calling above 'add_readers' procedure

```

```

drop procedure if exists delete_readers;

```

```

-- Below procedure delete many readers between two id numbers in 'czytelnicy' table

```

```

Delimiter \\\n
CREATE PROCEDURE delete_readers(nr_id1 INT,nr_id2 int)

BEGIN

label1: LOOP

    IF nr_id1 <= nr_id2 THEN

delete from czytelnicy where id =nr_id1;

SET nr_id1 = nr_id1 + 1;

        ITERATE label1;

    END IF;

    LEAVE label1;

END LOOP label1;

call readers();

END \\\n
delimiter ;

```

```

call delete_readers (11,113); -- Calling above 'delete_readers' procedure

```

```

drop procedure if exists update_readers;

```

```

-- Below procedure update reader

```

```

delimiter $$

CREATE PROCEDURE update_readers(nr_id INT,feature varchar(10), feature_data varchar(30))

begin

if feature = 'Name' then update czytelnicy set imie = feature_data where czytelnicy.id =nr_id;

```



```
elseif feature = 'Surname' then update czytelnicy set nazwisko = feature_data where czytelnicy.id =nr_id;

elseif feature = 'Dateofbirth' then update czytelnicy set data_urodzenia = feature_data where czytelnicy.id
=nr_id;

elseif feature = 'Gender' then update czytelnicy set plec = feature_data where czytelnicy.id =nr_id;

else select @val := 'Incorrect entered parameters!! Second parametr: Name Surname Dateofbirth Gender'
as Instruction_for_the_function;

end if;

    call readers();

end $$

delimiter ;

call update_readers(14,'Name','Karolina'); -- Calling above 'update_readers' procedure
```