

Курсове проектування з дисципліни:
Проектування високонавантажених систем зберігання даних

Тема:
Інформаційна система “Надання послуг типографії”

Керівник: Колесник Людмила Володимирівна
Виконав студент групи ІТКНУ-19-2
Марковець Назар Сергійович

Бізнес-процеси та їх бізнес-функції для різних типів користувачів

Незареєстрований користувач:

1. Робота з обліковими записами
 - 1.1. Реєстрація
2. Робота з каталогом послуг
 - 2.1. Перегляд усіх послуг
 - 2.2. Сортуння каталогу послуг
 - 2.3. Перегляд детальної інформації про послугу

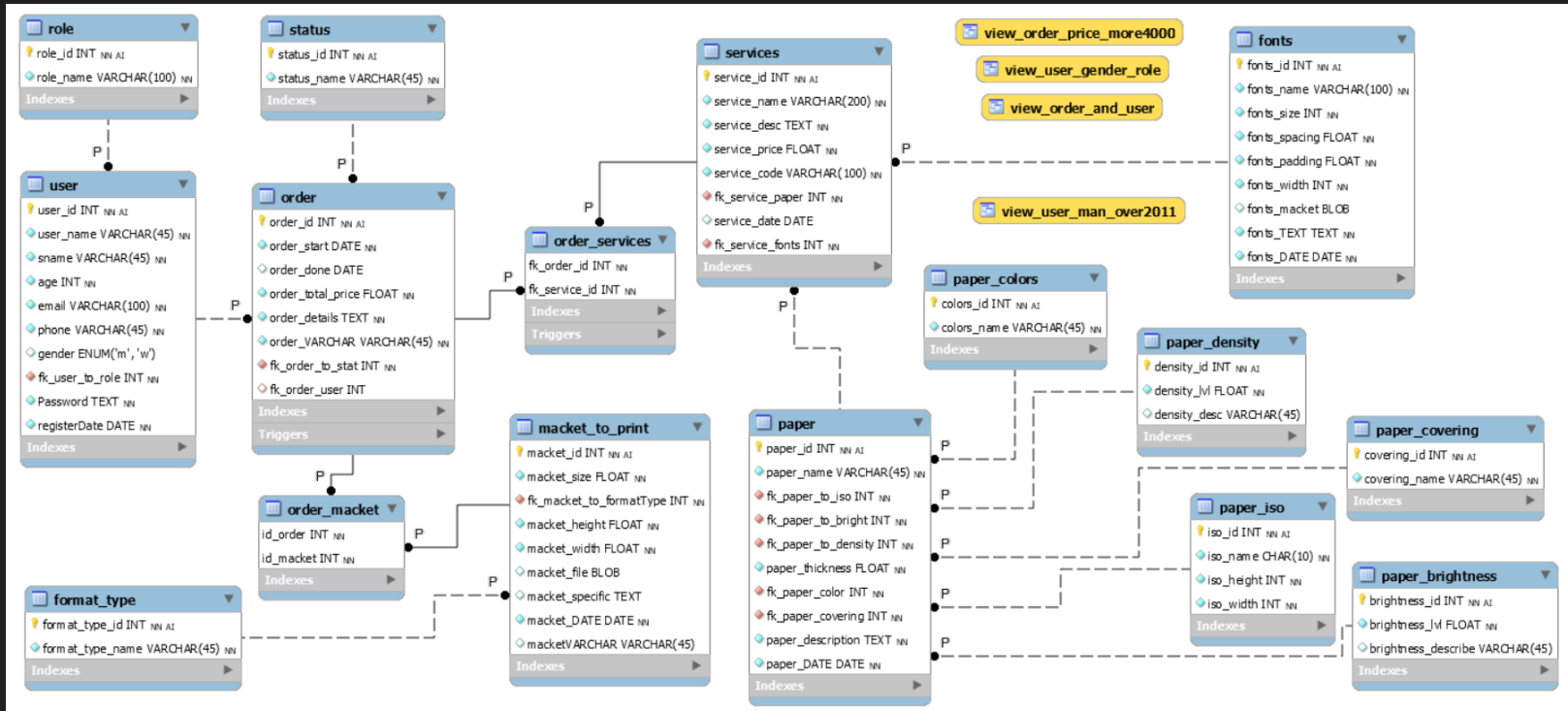
Зареєстрований користувач:

1. Робота з обліковими записами
 - 1.1. Вхід до системи
 - 1.2. Вихід з системи
2. Робота з каталогом послуг
 - 2.1. Перегляд усіх послуг
 - 2.2. Сортуння каталогу послуг
 - 2.3. Перегляд детальної інформації про послугу
3. Оформлення замовлення
 - 3.1 Додавання послуг до кошику
 - 3.2. Видалення послуг з кошику
 - 3.3. Підтвердження оформлення замовлення
4. Контроль замовлень
 - 4.1 Перегляд оформлених замовлень та їх статуси

Адміністратор:

1. Робота з обліковими записами
 - 1.1. Вхід до системи
 - 1.2. Вихід з системи
2. Робота з каталогом послуг
 - 2.1. Перегляд усіх послуг
 - 2.2. Сортуння каталогу послуг
 - 2.3. Перегляд детальної інформації про послугу
 - 2.4. Додавання нової послуги
 - 2.5. Редагування існуючої послуги
3. Робота з замовленнями
 - 3.1 Перегляд оформлених замовлень
 - 3.2. Зміна статусу замовлення
 - 3.3. Підтвердження оформлення замовлення
4. Контроль замовлень
 - 4.1 Перегляд оформлених замовлень та їх статуси

Фізична модель бази даних



Визначення посилальної цілісності

№	Ім'я таблиці 1, зовнішній ключ	Ім'я таблиці 2, первинний ключ	Тип посилальної цілісності для таблиці 1	Тип посилальної цілісності для таблиці 2	Тригер
1	User, fk_user_to_role	Role, role_id	ON INSERT RESTRICT	NO ACTION	Before insert
			ON UPDATE RESTRICT		Before update
			ON DELETE RESTRICT		Before delete
2	Order, fk_order_to_stat	Status, status_id	ON INSERT RESTRICT	NO ACTION	Before insert
			ON UPDATE RESTRICT		Before update
			ON DELETE RESTRICT		Before delete
3	Order, Fk_order_to_user	User, user_id	ON INSERT RESTRICT	NO ACTION	Before insert
			ON UPDATE RESTRICT		Before update
			ON DELETE RESTRICT	ON DELETE SET NULL	Before delete
4	Order_macket, id_order	Order, order_id	ON INSERT RESTRICT	NO ACTION	Before insert
			ON UPDATE RESTRICT		Before update
			ON DELETE RESTRICT		Before delete
5	Order_macket, Id_macket	Macket_to_print, macket_id	ON INSERT RESTRICT	NO ACTION	Before insert
			ON UPDATE RESTRICT		Before update
			ON DELETE RESTRICT		Before delete
6	Macket_to_print, fk_macket_to_formatType	Format_type, format_type_id	ON INSERT RESTRICT	NO ACTION	Before insert
			ON UPDATE RESTRICT		Before update
			ON DELETE RESTRICT	ON DELETE SET NULL	Before delete

SQL-код розроблених тригерів за специфікою високонавантаженої БД

```
CREATE DEFINER='root'@'localhost' TRIGGER `order_services_AFTER_INSERT` AFTER INSERT ON `order_services`  
FOR EACH ROW  
BEGIN  
    call my.update_total_price(new.fk_order_id);  
END
```

Тригер здійснює оновлення даних ціни замовлення після вставки нових даних до кошику клієнта

За замовчуванням при створенні замовлення ціна - 0

SQL-код процедури та функції, що викликаються

```
CREATE DEFINER='root'@'localhost' PROCEDURE `update_total_price`(in id int)  
BEGIN  
    update my.order set order_total_price = (select check_summ_for_service(id)) where order_id = id;  
END
```

```
CREATE DEFINER='root'@'localhost' FUNCTION `check_summ_for_service`(Ids int) RETURNS varchar(30) CHARSET utf8mb4  
DETERMINISTIC  
BEGIN  
  
    declare var varchar(30);  
    declare price varchar(30);  
    declare o_price float;  
    select order_total_price into o_price from my.order where order_id = Ids;  
    select sum(service_price) into var  
        from my.services  
        join my.order_services on fk_service_id = service_id  
        join my.order on order_id = fk_order_id  
        where order_id = Ids  
        group by fk_order_id  
        order by fk_order_id;  
    select if(var is null, o_price, var) into price;  
    RETURN price;  
END
```

SQL-код розроблених тригерів за специфікою високонавантаженої БД

```
DELIMITER $$
USE `my`$$
CREATE DEFINER = CURRENT_USER TRIGGER `my`.`order_BEFORE_INSERT_1` BEFORE INSERT ON `order` FOR EACH ROW
BEGIN
  if not exists (select 1 from my.`status` s where status_id = new.fk_order_to_stat)
  then
    signal sqlstate '45000' set message_text = 'Can not insert. There are no suitable status ';
  end if;
  if not exists (select 1 from my.`user` u where u.user_id = new.fk_order_user)
  then
    signal sqlstate '45000' set message_text = 'Can not insert. There are no suitable user ';
  end if;
END
$$
DELIMITER ;
```

Тригер здійснює перевірку відповідності даних з таблиці статусів замовлення та наявності користувача

SQL-код розроблених тригерів за специфікою високонавантаженої БД

```
CREATE DEFINER=`root`@`localhost` TRIGGER `user_BEFORE_DELETE` BEFORE DELETE ON `user` FOR EACH ROW BEGIN
  if exists (select 1 from my.`order` o where o.fk_order_user = old.user_id)
  then
    update my.`order` o
    set o.fk_order_user = null
    where o.fk_order_user = old.user_id;
  end if;
END
```

Тригер здійснює видалення ідентифікатора клієнта з замовлення.

SQL-код розроблених тригерів за специфікою високонавантаженої БД

```
CREATE DEFINER='root'@'localhost' TRIGGER `order_services_AFTER_DELETE` AFTER DELETE ON `order_services` FOR EACH ROW BEGIN
    call inno.uptate_total_price(old.fk_order_id);
END
```

Тригер здійснює оновлення даних ціни замовлення після видалення послуги з кошику клієнта

Створені уявлення

№	Назва	Призначення
1	view1	Представлення створене для відображення замовлень, розрахунку кінцевої ціни, виведення контактних даних користувача.
2	view2	Представлення сформоване на основі попереднього з вибіркою замовлень в яких ціна більша за 4000
3	view3	Створене представлення відображає дані про користувачів та їх ролі
4	view4	Будується на основі попереднього відбираючи дані чоловіків, зареєстрованих пізніше 2011-01-01 року.

```

1 • create or replace algorithm = merge
2 view view1 as
3 select
4     order_id,
5     order_start,
6     order_done,
7     check_summ_for_service(order_id),
8     fullname(user_name, sname) as User_name
9 from inno.order
10 join inno.user on user_id = fk_order_user;

```

#	Time	Action	Message	Duration / Fetch
1	13:05:43	create or replace algorithm = merge view vi...	0 row(s) affected	0.031 sec

```

1 • SELECT * FROM inno.view1;

```

order_id	order_start	order_done	check_sur	User_name	phone	email
1	2003-09-20	2003-09-23	1382	Lorenca Kashina	(+380)-35-763-6816	LorencaKashinagmail.com
2	2014-02-27	2014-03-02	380	Dzhuliana Zvarykina	(+380)-24-384-1519	DzhulianaZvarykinagmail.com
3	2002-03-16	2002-03-19	3007	Mirella Cherkasskaya	(+380)-97-114-6157	MirellaCherkasskayagmail.com
4	2006-12-18	2006-12-21	380	Alessiya Czimmerman	(+380)-63-391-9474	AlessiyaCzimmermangmail.com
5	2016-03-10	2016-03-13	2853	Kyara Annenkova	(+380)-37-984-9845	KyaraAnnenkovagmail.com
6	2019-05-16	2019-05-19	2521	Violetta Eroplina	(+380)-25-788-7159	ViolettaEroplinagmail.com
7	2019-11-01	2019-11-04	2643	Dzhovanna Stoczka	(+380)-18-598-9891	DzhovannaStoczkaagmail.com
8	2014-06-03	2014-06-06	1393	Patricziya Vodynskaya	(+380)-63-797-5969	PatricziyaVodynskayagmail.com
9	2006-12-03	2006-12-06	3088	Albina Zubatova	(+380)-33-449-4635	AlbinaZubatovagmail.com
10	2012-08-13	2012-08-16	1943	Marianna Paleczkaya	(+380)-88-265-5516	MariannaPaleczkayagmail.com
11	2013-10-21	2013-10-24	460	Aureliana Davydova	(+380)-76-243-5283	AurelianaDavydovagmail.com
12	2020-09-04	2020-09-07	3518	Mariya Loganova	(+380)-26-697-9269	Mariyaloganovagmail.com
13	2012-12-04	2012-12-07	1746	Mirella Cherkasskaya	(+380)-97-114-6157	MirellaCherkasskayagmail.com
15	2005-03-15	2005-03-18	2084	Daniela Korolenko	(+380)-37-586-8499	DanielaKorolenkogmail.com

View1

```

1 • create or replace algorithm = merge
2 view view2 as
3 select
4     *
5 from inno.view1
6 where price > 4000;
7

```

#	Time	Action	Message	Duration / Fetch
1	14:00:40	create or replace algorithm = merge view view2 as select * ...	0 row(s) affected	0.031 sec

```

1 • SELECT * FROM inno.view2;

```

order_id	order_start	order_done	price	User_name	phone	email
47	2018-02-27	2018-03-02	4592	Margarita Moiseeva	(+380)-66-366-2312	MargaritaMoiseevagmail.com
1580	2012-04-08	2012-04-11	4495	Gavin Booth	(+380)-93-259-9521	GavinBooth7gmail.com
1619	2004-08-26	2004-08-29	4496	Brian Small	(+380)-76-193-1473	BrianSmall7gmail.com
1865	2005-07-24	2005-07-27	4078	Colleen Boone	(+380)-34-585-2227	ColleenBoone2gmail.com
1956	2008-06-17	2008-06-20	4342	John Boyd	(+380)-92-848-3146	JohnBoyd6gmail.com
2077	2009-03-24	2009-03-27	4460	Charles Garrett	(+380)-69-379-6391	CharlesGarrett5gmail.com
2368	2018-07-01	2018-07-04	4206	Julius Hampton	(+380)-38-872-5818	JuliusHampton8gmail.com
2682	2000-09-11	2000-09-14	4404	Richard Wiggins	(+380)-61-216-3471	RichardWiggins3gmail.com
2724	2001-09-10	2001-09-13	4195	Lawrence Dorsey	(+380)-73-495-2516	LawrenceDorsey9gmail.com
2774	2012-06-19	2012-06-22	4041	Katherine Chase	(+380)-62-963-7587	KatherineChase8gmail.com
3446	2020-07-08	2020-07-11	4440	Steven McKinney	(+380)-89-753-1678	StevenMcKinney3gmail.com
3690	2018-08-26	2018-08-29	4195	Peter Wells	(+380)-88-146-5313	PeterWells3gmail.com

View2

View3

```

1 • create or replace algorithm = temptable
2 view view3 as
3 select
4     user_id,
5     fullname(user_name, sname) as `Fullname`,
6     age, email,
7     phone,
8     if(gender > 0, 'Men', 'Women') as gender,
9     registerDate,
10    role_name
11
12 from inno.user
13 join inno.role on role_id = fk_user_to_role;
14
15 • select * from inno.view3

```

user_id	Fullname	age	email	phone	gender	registerDate	role_name
1	Sara Stroynovskaya	55	SaraStroynovskayagmail.com	(+380)-66-689-3517	Women	2011-04-08	admin
2	Enriketta Goronovich	16	EnrikettaGoronovichgmail.com	(+380)-43-981-7661	Men	2011-02-23	user
3	Kyara Annenkova	35	KyaraAnnenkovagmail.com	(+380)-37-984-9845	Men	2005-09-24	user
4	Valentina Vishneveczkaya	24	ValentinaVishneveczkayagmail.com	(+380)-68-856-3674	Men	2012-04-09	user
5	Dzheraldina Bolkunova	86	DzheraldinaBolkunovagmail.com	(+380)-71-537-5156	Men	2000-08-08	user
6	Albina Buturlina	57	AlbinaButurlinagmail.com	(+380)-16-474-6224	Women	2013-07-23	user
7	Marianna Paleczkaya	53	MariannaPaleczkayagmail.com	(+380)-88-265-5516	Women	2019-07-21	user
8	Leticziya Steczkaya	65	LeticziyaSteczayagmail.com	(+380)-64-113-9233	Women	2015-06-12	user
9	Veronika Kuzmina	48	VeronikaKuzminagmail.com	(+380)-61-928-7576	Men	2016-04-22	user

View4

```

1 • create or replace algorithm = temptable
2 view view4 as
3 select
4     user_id,
5     Fullname,
6     age, email,
7     phone,
8     gender,
9     registerDate,
10    role_name
11
12 from inno.view3
13 where gender like 'Men' and registerDate > '2011-01-01';
14 • select * from inno.view4
15
16

```

user_id	Fullname	age	email	phone	gender	registerDate	role_name
2	Enriketta Goronovich	16	EnrikettaGoronovichgmail.com	(+380)-43-981-7661	Men	2011-02-23	user
4	Valentina Vishneveczkaya	24	ValentinaVishneveczkayagmail.com	(+380)-68-856-3674	Men	2012-04-09	user
9	Veronika Kuzmina	48	VeronikaKuzminagmail.com	(+380)-61-928-7576	Men	2016-04-22	user
10	Albina Zubatova	76	AlbinaZubatovagmail.com	(+380)-33-449-4635	Men	2019-05-17	user
12	Virdzhiniya Sibirskaya	44	VirdzhiniyaSibirskayagmail.com	(+380)-35-144-6976	Men	2015-02-05	user
13	Rossana Ostrozhskaya	23	RossanaOstrozhskayagmail.com	(+380)-41-257-5513	Men	2019-12-09	user
14	Gabriella Afanassenko	92	GabriellaAfanassenkogmail.com	(+380)-99-545-4677	Men	2020-08-13	user

Розробка уявлень

```
CREATE DEFINER='root'@'localhost' PROCEDURE `get_paper_covering_REPEAT`()
BEGIN
    declare v_paper_id, v_fk_paper_covering, count_row, i int;
    declare v_paper_name varchar(30);
    declare c1 cursor for Select paper_id, paper_name, fk_paper_covering from inno.paper;

    drop table if exists `tmp_paper`;
    CREATE TEMPORARY TABLE `tmp_paper`
    (
        `id` int unsigned NOT NULL AUTO_INCREMENT, `paper_name` varchar(255) NOT NULL, `covering` varchar(255) NOT NULL, PRIMARY KEY (`id`)
    )
    ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO_INCREMENT=1;
    select count(1) into count_row from inno.paper;
    open c1; set i = 1;
    repeat
        fetch c1 into v_paper_id, v_paper_name, v_fk_paper_covering;
        if v_fk_paper_covering = 1 then insert into `tmp_paper` values ( v_paper_id, v_paper_name, 'none');
        elseif v_fk_paper_covering = 2 then insert into `tmp_paper` values ( v_paper_id, v_paper_name, 'matt');
        elseif v_fk_paper_covering = 3 then insert into `tmp_paper` values ( v_paper_id, v_paper_name, 'shoth');
        elseif v_fk_paper_covering = 4 then insert into `tmp_paper` values ( v_paper_id, v_paper_name, 'rougt');
        elseif v_fk_paper_covering = 5 then insert into `tmp_paper` values ( v_paper_id, v_paper_name, 'glossy');
        end if;
        set i = i+1;
    until (i > count_row)
    END repeat;
    close c1;
    select * from inno.`tmp_paper`;
    drop table if exists `tmp_paper`;
END
```

Після виклику даної процедури користувач отримає дані про назву паперу та тип покриття використовуючи оператор циклу REPEAT.

```
1 call get_paper_covering_REPEAT();
```

Result Grid			
Filter Rows:			
Exports			
	id	paper_name	covering
1	BR92		glossy
2	^Q1246		none
3	MH1437		glossy
4	DQ1417		rougt
5	YA1072		glossy
6	IH574		matt
7	ZS1293		glossy
8	GH802		matt
9	SK58		rougt
10	GL1		none
11	RD103		glossy
12	TG1953		none
13	_X1337		none
14	X\455		glossy

Розробка функцій

```
CREATE DEFINER='root'@'localhost' FUNCTION `fw_get_maket_format`(var smallint) RETURNS varchar(400) CHARSET utf8mb4
DETERMINISTIC
BEGIN
declare `this_maket_format` varchar(400) default null;
select
concat
(
'Maket №: ', maket_id, ', size MB: ', maket_size,
', use format - ',
format_type_name, ', parameters: ',
maket_height, ' pxl x ',
maket_width, ' pxl'
) into `this_maket_format`
from inno.maket_to_print, inno.format_type
where fk_maket_to_formatType = format_type_id and maket_id = var;

RETURN `this_maket_format`;
END
```

Вивести макети для друку, що надають споживачі послуг та їх формат.

```
1 • select maket_id, fw_get_maket_format(maket_id) from inno.maket_to_print;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	maket_id	fw_get_maket_format(maket_id)			
▶	4	Maket №: 4, size MB: 4.1, use format - .GIF, parameters: 1048.8 pxl x 2170.6 pxl			
	5	Maket №: 5, size MB: 7.1, use format - .GIF, parameters: 712.7 pxl x 3482.8 pxl			
	14	Maket №: 14, size MB: 41.4, use format - .GIF, parameters: 1846.5 pxl x 2041.6 pxl			

Створення транзакцій

№ Призначення транзакції

1 Оформлення замовлення незареєстрованим клієнтом:

- Занесення інформації користувача до таблиці клієнтів
- Занесення інформації таблиці замовлень
- Занесення інформації до кошику

2 Зміна статусу замовлення

3 Додавання нової послуги

Найменування

таблиць

inno.user, inno.order,
order_services

inno.order, inno.status

inno.services,

inno.paper, inno.fonts

```

CREATE DEFINER='root'@'localhost' PROCEDURE `trans_register_user_create_order`()
BEGIN
  declare exit handler for sqlexception
  BEGIN
    ROLLBACK;
  END;
  start transaction;

  INSERT INTO `inno`.`user`
  VALUES (4002,'Oleg', 'Majami', '20', 'oleg.jama@outlook.com',
  '(+380)-44-323-3355', '1', '2', 'clscjsfoejfewofhkewflwhflwkjmlkc.jvsjdv',
  '2020-11-19');

  INSERT INTO `inno`.`order`
  VALUES ('4002', '2020-11-19', null, '0', 'test', 'test', '1', '4002');

  insert into `inno`.`order_services` values(4002, 5000);

  update inno.order set order_total_price = check_summ_for_service(4002);
  commit;
END

```

2 • `call trans_register_user();`

Output			
Action Output			
#	Time	Action	Message
1	00:06:31	call trans_register_user()	0 row(s) affected

1 • `SELECT * FROM inno.user where user_id = 4002;`

	user_id	user_name	sname	age	email	phone	gender	fk_user_to_role	History	registerDate
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Оскільки введено невірні дані — відбувається rollback транзакції і до кошику не потрапляє жодна послуга. Після створення процедури з порушеною транзакцією можна протестувати її викликом та оглянути результати.

Виправлення введених даних

```
CREATE DEFINER='root'@'localhost' PROCEDURE `trans_register_user_create_order`()
BEGIN
  declare exit handler for sqlexception
  BEGIN
    ROLLBACK;
  END;
  start transaction;

  INSERT INTO `inno`.`user`
  VALUES (4002, 'Oleg', 'Majami', '20', 'oleg.jama@outlook.com',
  '(+380)-44-323-3355', '1', '2', 'clscjsfoejfewfhkewflwhflwkjmlkc.jvsjdv',
  '2020-11-19');

  INSERT INTO `inno`.`order`
  VALUES ('4002', '2020-11-19', null, '0', 'test', 'test', '1', '4002');

  insert into `inno`.`order_services` values(4002, 3994);

  update inno.order set order_total_price = check_summ_for_service(4002);
  commit;
END
```

```
1
2 • call trans_register_user();
```

Output

Action Output

#	Time	Action	Message
1	00:12:05	Apply changes to trans_register_user_create_order	No changes detected
2	00:12:56	Apply changes to trans_register_user_create_order	No changes detected
3	00:13:03	call trans_register_user()	0 row(s) affected

```
1 • SELECT * FROM inno.order_services where fk_order_id = 4002;
```

<

Result Grid



Filter Rows:

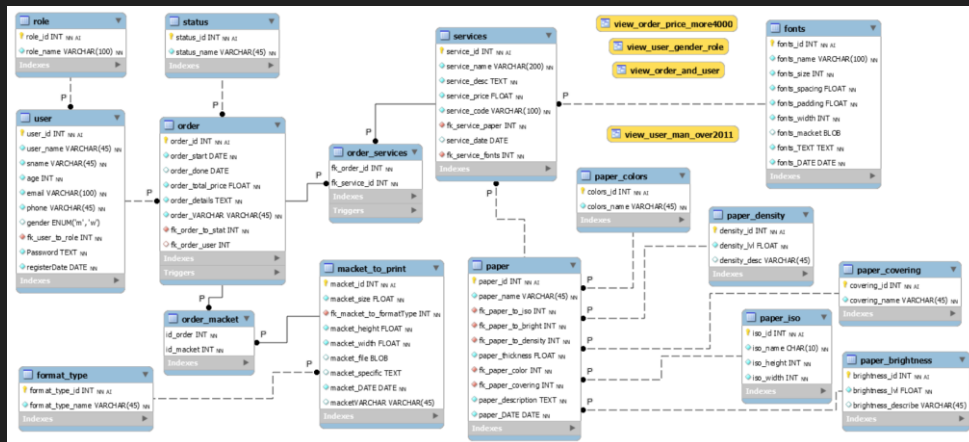
Edit:



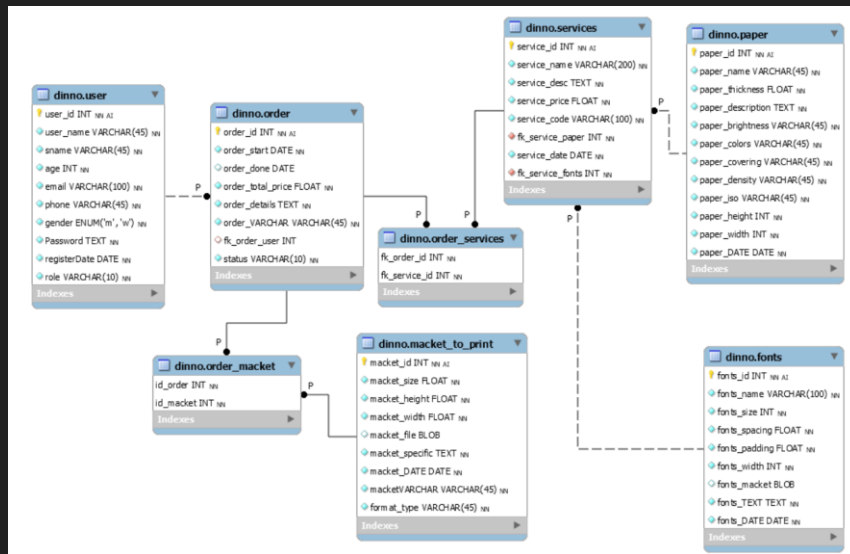
Export/Import:

	fk_order_id	fk_service_id
▶	4002	3999
*	NULL	NULL

Фізична модель до денормалізації



Фізична модель після денормалізації



Порівняння високонавантажених SQL-запитів для нормалізованої та денормалізованої БД

```
2 • select
3     service_id,
4     service_name,
5     service_price,
6     fonts_name,
7     fonts_padding,
8     fonts_size,
9     paper_name,
10    paper_thickness,
11    iso_name,
12    iso_height,
13    iso_width,
14    covering_name,
15    colors_name,
16    service_desc
17 from inno.services
18 join inno.paper on paper_id = fk_service_paper
19 join inno.fonts on fonts_id = fk_service_fonts
20 join paper_colors on colors_id = fk_paper_color
21 join paper_iso on iso_id = fk_paper_to_iso
22 join paper_covering on covering_id = fk_paper_covering
23 where service_name like 'AX';
24
```

Query Statistics

Timing (as measured at client side):
Execution time: 0:00:0.0310000

Вивести детальні дані про послугу

```
1 • use dinno;
2 • select
3     service_id,
4     service_name,
5     service_price,
6     fonts_name,
7     fonts_padding,
8     fonts_size,
9     paper_name,
10    paper_thickness,
11    paper_iso,
12    paper_height,
13    paper_width,
14    paper_colors,
15    paper_covering
16 from dinno.services
17 join dinno.paper on paper_id = fk_service_paper
18 join dinno.fonts on fonts_id = fk_service_fonts
19 where service_name like 'AX';
20
21
```

Query Statistics

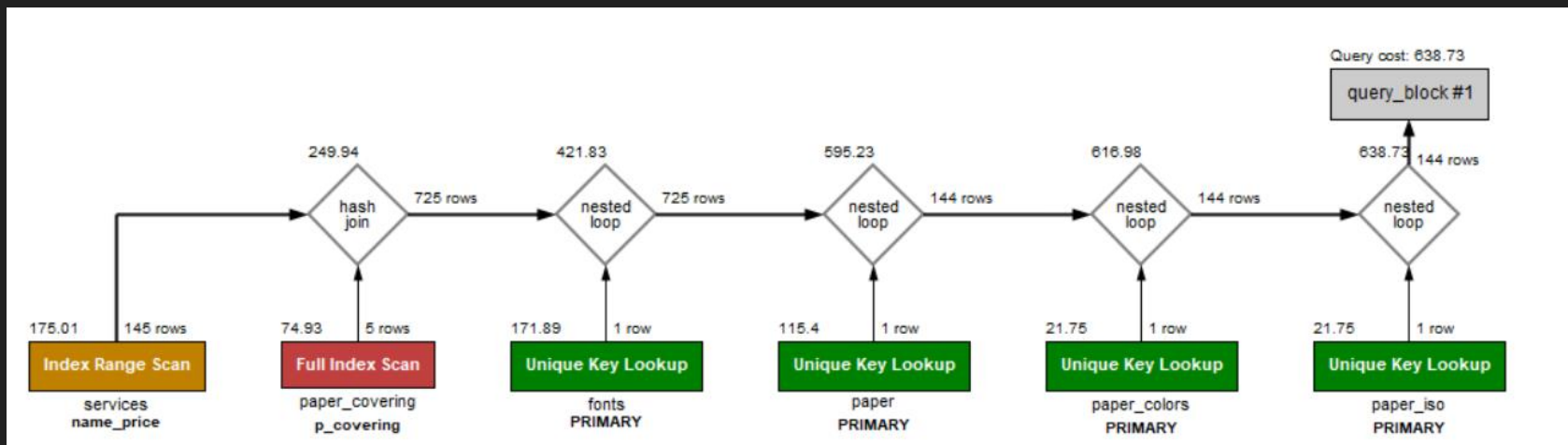
Timing (as measured at client side):
Execution time: 0:00:0.0780000

Запит до нормалізованої бази даних

	service_id	service_name	service_price	fonts_name	fonts_padding	fonts_size	paper_name	paper_thickness	paper_iso	paper_height	paper_width	paper_colors	paper_covering
▶	4001	Light correction	250	IIWW3	2.5	14	NN332	20	A1	594	841	green	glorry
	4000	\^1446	784	[UOHKO	3.4	7	EE1461	4.8	B0	1000	1414	yellow	none
	3999	I]1448	642	^UQYBS	1.1	16	DA933	1.6	A0	841	1189	blue	matt
	3998	IC977	635	Z]BTLK	2.9	16	SL787	4.4	B3	353	500	dark gray	glorry
	3997	EL717	874	SZ]SHO	3.3	18	W\1280	4.6	C5	162	229	black	smooth
	3996	UQ1877	634	EEMDKB	3.9	19	TU512	4.5	C8	57	81	white	glorry
	3995	JJ1413	721	Gallard BT	2.7	7	BA489	4.9	B9	44	62	black	glorry
	3994	A[934	576	ChelthmITC Bk BT	2.6	21	HF336	2.3	A8	52	74	black	rough
	3993	YP1580	626	Antique Olive	3.3	7	[X1126	1.7	A4	210	297	blue	smooth
	3992	OR1823	922	OJ_FRG	3.9	17	WP279	2.9	B4	250	353	dark gray	none

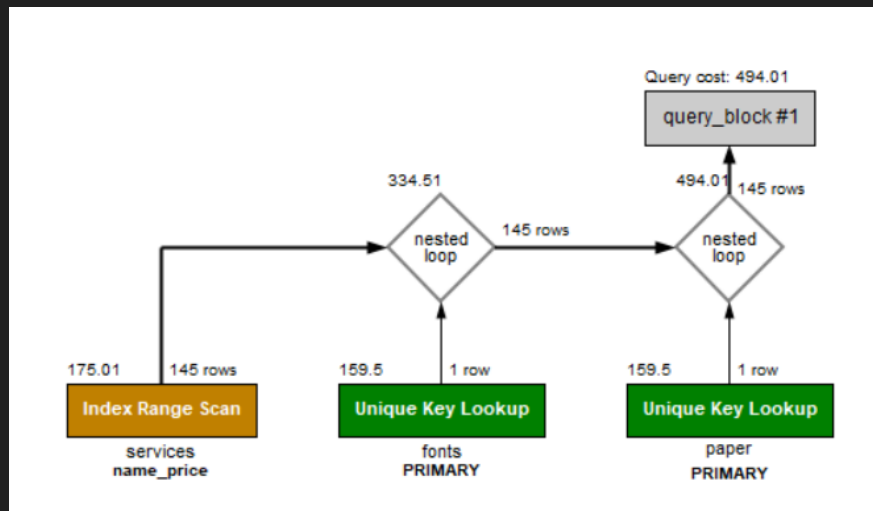
Запит до денормалізованої бази даних

План виконання запиту до денормалізації



Tabular Explain											
id	select_type	table	partitions	type	possible_keys	key	key...	ref	rows	filtered	Extra
1	SIMPLE	services		range	fk_service_paper...	name_price	14		145	100.00	Using where
1	SIMPLE	paper_covering		index	PRIMARY	p_covering	137		5	100.00	Using index; Using join buffer (hash join)
1	SIMPLE	fonts		eq_ref	PRIMARY	PRIMARY	4	inno.services.fk_service_fonts	1	100.00	
1	SIMPLE	paper		eq_ref	PRIMARY,fk_pap...	PRIMARY	4	inno.services.fk_service_paper	1	20.00	Using where
1	SIMPLE	paper_colors		eq_ref	PRIMARY	PRIMARY	4	inno.paper.fk_paper_color	1	100.00	
1	SIMPLE	paper_iso		eq_ref	PRIMARY	PRIMARY	4	inno.paper.fk_paper_to_iso	1	100.00	

План виконання запиту після денормалізації



id	select_type	table	partitions	type	possible_keys	key	key...	ref	rows	filtered	Extra
1	SIMPLE	services		range	fk_service_paper...	name_price	14		145	100.00	Using where
1	SIMPLE	fonts		eq_ref	PRIMARY	PRIMARY	4	dinno.services.fk_service_fonts	1	100.00	
1	SIMPLE	paper		eq_ref	PRIMARY	PRIMARY	4	dinno.services.fk_service_paper	1	100.00	

Реалізація складеного індексу

```
1 • select
2   user_id,
3   user_name,
4   sname,
5   age,
6   email,
7   phone,
8   gender,
9   registerDate
10  from inno.user
11  where user_name = 'Sara';
```

Tabular Explain											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user		ref	name_age	name_age	14	const	9	100.00	Using where

```
1 • select
2   user_id,
3   user_name,
4   sname,
5   age,
6   email,
7   phone,
8   gender,
9   registerDate
10  from inno.user
11  where user_name = 'Sara';
```

Tabular Explain											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	user		ALL					3919	10.00	Using where

```
ALTER TABLE `inno`.`user`
ADD INDEX `name_age` (`user_name`(4) ASC, `age` ASC) VISIBLE;
;
```

Запит з індексом

Timing (as measured at client side):

Execution time: 0:00:0.00000000

Timing (as measured by the server):

Execution time: 0:00:0.00035630

Table lock wait time: 0:00:0.00010800

Запит без індексу

Timing (as measured at client side):

Execution time: 0:00:0.01600000

Timing (as measured by the server):

Execution time: 0:00:0.00238030

Table lock wait time: 0:00:0.00015400

FULLTEXT індекси

```
CREATE FULLTEXT INDEX `macket_text` on inno.macket_to_print (`macket_specific`, `macketVARCHAR`);
```

```
1 • select distinct
2   macket_id,
3   concat(macket_size, ' ', 'Mb') as size,
4   concat(macket_height, ' px') as height,
5   concat(macket_width, ' px') as width,
6   macket_DATE,
7   macket_specific,
8   match(macket_specific,macketVARCHAR) against ('Африку побережье' IN NATURAL LANGUAGE MODE) as coin
9   from inno.macket_to_print
10  where match(macket_specific,macketVARCHAR) against ('Африку побережье' IN NATURAL LANGUAGE MODE);
11
```

	macket_id	size	height	width	macket_DATE	macket_specific	coin
▶	650	12.8 Mb	2726.4 px	3021.7 px	2010-04-14	Карта 5. Набег викингів на Южную Европу и Северную Африку в 829-8 61 гг...	9.383395195007324
	33	34.1 Mb	507.9 px	1193.9 px	2009-09-13	Мчалась стремительноКарта 5. Набег викингів на Южную Европу и Северную...	6.429466724395752
	187	29.2 Mb	2810.2 px	1088.2 px	2009-01-06	на дереве тонКарта 5. Набег викингів на Южную Европу и Северную Африк...	6.429466724395752
	492	15.3 Mb	3145.7 px	3579.3 px	2009-11-07	Разгромив отряды герцога Бордоского и Сентокого, Сегуина, который пал в с...	6.429466724395752
	962	25.8 Mb	626.1 px	1776.3 px	2011-02-20	Карта 5. Набег викингів на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	1684	6.2 Mb	3650.3 px	1283.2 px	2013-02-11	Карта 5. Набег викингів на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	2023	26.3 Mb	1904.9 px	2933.7 px	2014-01-16	Карта 5. Набег викингів на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	2030	15.1 Mb	3788.1 px	3676.2 px	2014-01-23	Карта 5. Набег викингів на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	2136	10.3 Mb	3428.4 px	3938.7 px	2014-05-09	Вокруг в Исландию устремились многочисленные переселенцы, поток которых...	6.429466724395752
	2790	47.1 Mb	1349.1 px	3591.4 px	2016-02-22	«Житие святого Ансагрия»[39] сообщает об этом набеге так: «Выпал жребий, ...	6.429466724395752
	3739	18.9 Mb	1134.5 px	974.6 px	2018-09-28	С наступлением тепла норманны покинули свой лагерь и с богатой добычей от...	6.429466724395752
	50	37.8 Mb	1081.2 px	2686.9 px	2008-08-22	Норманны ограбили побережье Тосканы, разорили город Пизу и ряд других пр...	4.430892467498779
	2432	37.3 Mb	675.6 px	2991.8 px	2015-03-01	Норманны ограбили побережье Тосканы, разорили город Пизу и ряд других пр...	4.430892467498779
	161	18.8 Mb	2092.4 px	3978.8 px	2008-12-11	То ли нуждаясь во времени для «переваривания» обильной добычи, то ли в си...	2.9539284706115723
	296	10.9 Mb	883.8 px	1514.1 px	2009-04-25	Прочность морских рубежей державы франков вынудила норманнов искать уд...	2.9539284706115723
	690	41.5 Mb	957.3 px	1623.4 px	2010-05-24	На принадлежность подкурганных ладейных кренашей и ингунашей именно ви...	2.9539284706115723
	1182	19.5 Mb	3040.1 px	3836.5 px	2011-09-28	Поконив с Савреной, Олав Толстый направил флотилию к юго-западному поб...	2.9539284706115723
	1232	21.9 Mb	1364.7 px	2191.8 px	2011-11-17	Проведя на побережье Восточной Англии почти год, «Великое войско», — как ...	2.9539284706115723

Повернено 240 рядків

Tabular Explain											
id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	macket_to_print		fulltext	macket_text	macket_text	0	const	1	100.00	Using where; Ft_hints: sorted

Знайти рядки що містять слово «Африку», розміщення вище, якщо присутнє слово «побережье»

```
1 • select distinct
2   macket_id,
3   concat(macket_size, ' ', 'Mb') as size,
4   concat(macket_height, ' px') as height,
5   concat(macket_width, ' px') as width,
6   macket_DATE,
7   macket_specific,
8   match(macket_specific,macketVARCHAR) against ('Африку - побережье' IN BOOLEAN MODE) as coin
9   from inno.macket_to_print
10  where match(macket_specific,macketVARCHAR) against ('+Африку -побережье' IN BOOLEAN MODE);
```

Result Grid							
Filter Rows:							
Export: Wrap Cell Content:							
	macket_id	size	height	width	macket_DATE	macket_specific	coin
▶	33	34.1 Mb	507.9 px	1193.9 px	2009-09-13	Мчалась стремительноКарта 5. Набеги викингов на Южную Европу и Северну...	6.429466724395752
	187	29.2 Mb	2810.2 px	1088.2 px	2009-01-06	на дереве том,Карта 5. Набеги викингов на Южную Европу и Северную Африк...	6.429466724395752
	492	15.3 Mb	3145.7 px	3579.3 px	2009-11-07	Разгромив отряды герцога Бордосского и Сентокого, Сегуина, который пал в с...	6.429466724395752
	962	25.8 Mb	626.1 px	1776.3 px	2011-02-20	Карта 5. Набеги викингов на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	1684	6.2 Mb	3650.3 px	1283.2 px	2013-02-11	Карта 5. Набеги викингов на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	2023	26.3 Mb	1904.9 px	2933.7 px	2014-01-16	Карта 5. Набеги викингов на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	2030	15.1 Mb	3788.1 px	3676.2 px	2014-01-23	Карта 5. Набеги викингов на Южную Европу и Северную Африку в 829-8 61 гг...	6.429466724395752
	2136	10.3 Mb	3428.4 px	3938.7 px	2014-05-09	Воюере в Исландию устремились многочисленные переселенцы, поток которых...	6.429466724395752
	2790	47.1 Mb	1349.1 px	3591.4 px	2016-02-22	«Житие святого Ансагрия»[39] сообщает об этом набеге так: «Выпал жребий, ...	6.429466724395752
	3739	18.9 Mb	1134.5 px	974.6 px	2018-09-28	С наступлением тепла норманны покинули свой лагерь и с богатой добычей от...	6.429466724395752

Повернено 11 рядків

```

1 • select distinct
2   macket_id,
3   concat(macket_size, ' ', 'Mb') as size,
4   concat(macket_height, ' px') as height,
5   concat(macket_width, ' px') as width,
6   macket_DATE,
7   macket_specific,
8   match(macket_specific,macketVARCHAR) against ('Африку побережье' WITH QUERY EXPANSION) as coin
9   from inno.macket_to_print
10  where match(macket_specific,macketVARCHAR) against ('Африку побережье' WITH QUERY EXPANSION);

```

Result Grid								Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	macket_id	size	height	width	macket_DATE	macket_specific	coin				
▶	2621	42.8 Mb	1875.9 px	3075.4 px	2015-09-06	Муспель излучал в Мировую Бездну тепло и свет, а Нифльхейм дышал свире...	1219.548583984375				
	95	20.4 Mb	3807.2 px	3077.2 px	2008-10-06	Муспель излучал в Мировую Бездну тепло и свет, а Нифльхейм дышал свире...	1193.3900146484375				
	701	25.9 Mb	1562.7 px	3141.3 px	2010-06-04	Муспель излучал в Мировую Бездну тепло и свет, а Нифльхейм дышал свире...	1186.2064208984375				
	3547	13.6 Mb	2085.3 px	1901.6 px	2018-03-20	Превзошедший все ожидания успех плавания принес Лейву Эйриксону прозви...	1137.8624267578125				
	3918	22.9 Mb	2965.5 px	2804.4 px	2019-03-26	Муспель излучал в Мировую Бездну тепло и свет, а Нифльхейм дышал свире...	1133.1806640625				
	2748	14.1 Mb	654.5 px	2998.4 px	2016-01-11	Затем братья отправились в лес и из приглянувшихся деревьев создали мужчи...	1131.9984130859375				
	2922	37.4 Mb	1663.5 px	2357.2 px	2016-07-03	Муспель излучал в Мировую Бездну тепло и свет, а Нифльхейм дышал свире...	1109.76513671875				
	3249	47.4 Mb	3685.6 px	978.8 px	2017-05-26	Через 12 лет после резни в Линнерике у холма Тара (Лара, северо-западнее Д...	1102.9366455078125				
	1240	37.6 Mb	1385.6 px	2807.4 px	2011-11-25	Муспель излучал в Мировую Бездну тепло и свет, а Нифльхейм дышал свире...	1093.4002685546875				
	1621	22.2 Mb	3430.8 px	3933.1 px	2012-12-10	При изготовлении наконечников немалое внимание уделялось декоративной о...	931.2218017578125				
	3323	34.8 Mb	3713.5 px	821.4 px	2017-08-08	Корабли викингов, несомненно, в значительной степени оказались бы лишены...	854.6871337890625				
	70	15.1 Mb	1993.1 px	593.3 px	2008-09-11	В отличие от Запада, где интерес к деяния скандинавских воинов-мореходо...	852.3653564453125				
	2290	15.9 Mb	3342.2 px	3878.4 px	2014-10-10	Вместе с конунгом Олавом Гудредсоном в курган близ Гокстада последовали 1...	833.8034057617188				
	1222	25.4 Mb	3852.2 px	3808.7 px	2011-11-07	Покончив с Сааремой, Олав Толстый направил флотилию к юго-западному поб...	806.8876342773438				
	2499	42.9 Mb	3405.7 px	2774.6 px	2015-05-07	В начале IX века «каролингские» мечи викингов имеют прямые перекрестья и ...	800.564697265625				
	3849	38.7 Mb	2254.9 px	2434.8 px	2019-01-16	Без особого труда захватив и разграбив Себорг, шведы подожгли его и устре...	799.5993041992188				
	2654	32.4 Mb	2685.6 px	2440.8 px	2015-10-09	Однако угроза лишь подтолкнула англичан к действию. В Нормандию отпра...	789.7273559570312				

Повернено 3997 рядків

Висновки

Практичним шляхом було виявлено, що денормалізація збільшує приріст виконання запиту лише у деяких випадках. В більшості випадків запити до денормалізованої бази даних виконувались з такою ж швидкістю, а іноді повільніше, и лише в поодиноких випадках в два рази швидше.

В кінцевому випадку варто використовувати нормалізовану базу даних використовуючи InnoDB в третій нормальній формі з причини високої необхідності не тільки вибірки даних, а й записів. У випадку виникнення аномалій видалення, невірної вставки тощо – ресурсні втрати будуть суттєвими.

Реалізація посилювальної цілісності для MyISAM за допомогою тригерів знижує швидкість розробки та в деяких випадках швидкість виконання запиту.

Для повнотекстового пошуку послуг, або макетів за описом, назвою обрано NATURAL LANGUAGE MODE за найбільшу точність пошукових результатів.