POLITECHNIKA GDAŃSKA

Optymalizacja baz danych

Report 1

Oskar Kwidziński

S156013

Informatyka Stosowana

Wydział Fizyki Technicznej i Matematyki Stosowanej

In accordance with commands to Report 1 a database has been created. The project was created in Oracle database managment system (version 11g) with Oracle SQLDeveloper as a data modeler. It consists of 13 entities and its main purpose is to help manage an online shop with collecting and processing data about employees, sales, supplieres, customers etc. The ER diagram of database is presented in Fig. 1.

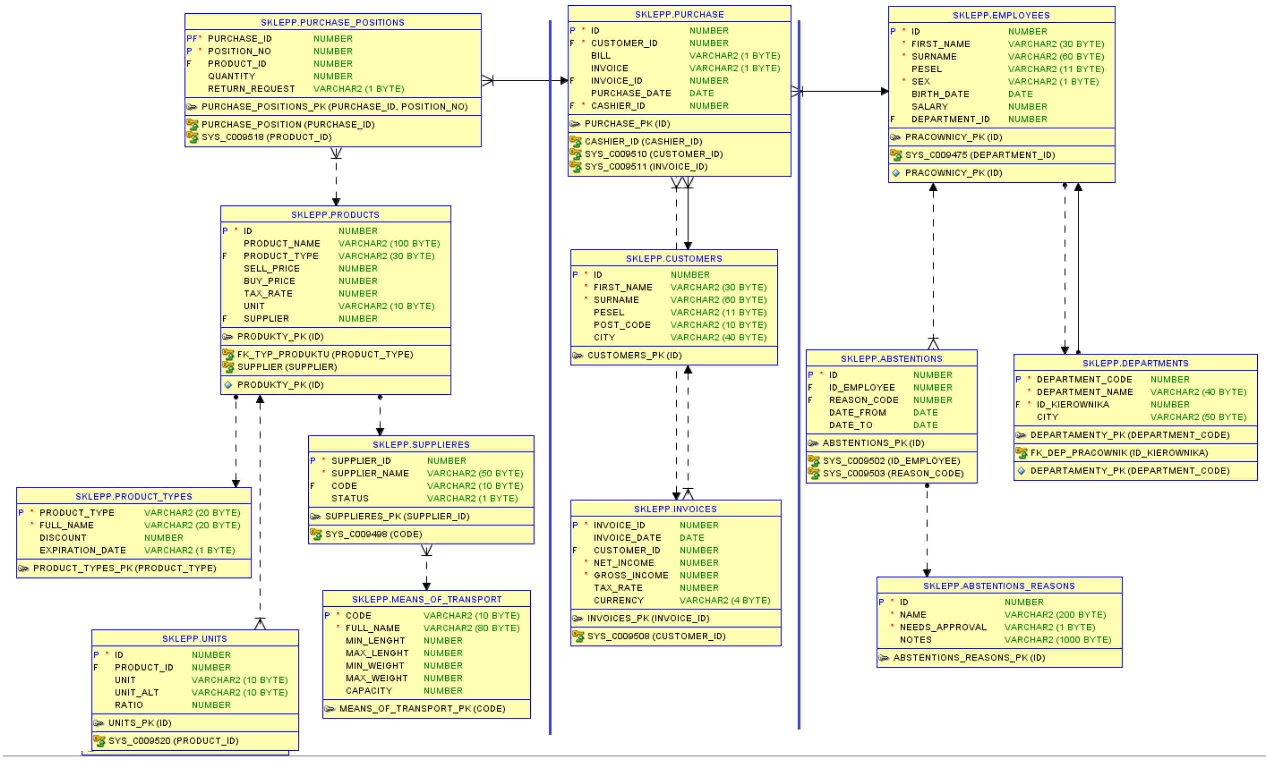


Fig. 1. Schema of the database.

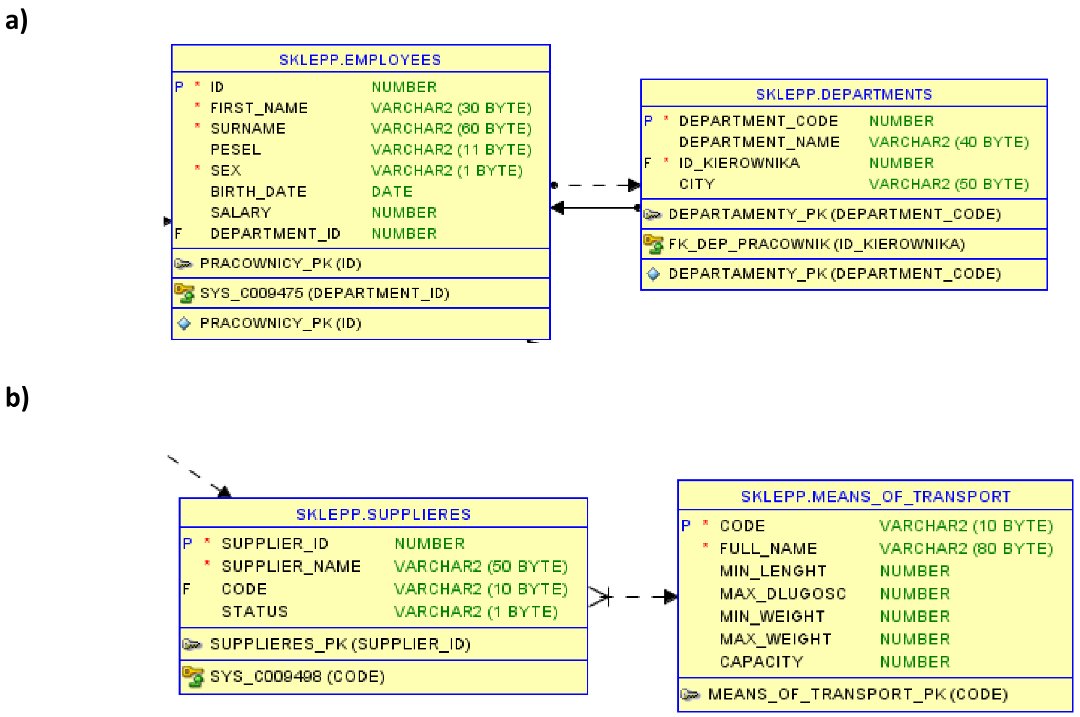
The main task for which the base has been created was to create an environment for managing the entire enterprise. ER diagram in Fig. 1. presents branches of database, that can be devided in three groups:

- Left: *assortment and stock* - providing information about products sold in shop. To help managing orders and purchased goods ‘Purchase positions’ table has been created. It is associeted with ‘Purchase’ table, however its desire is to push information about products in stock to ‘Products’ table. Managment with all the products details can be accomplished through all encies placed below.

- Middle: *purchase processing* – consists of customers personal data ‘Customers’, list of products one has purchased ‘Purchase’ and optionally an invoice requests ’Invoices’.

- Right: *employees data* – collection performing a role of register for all employees: personal data stored in ‘Employees’, association with department ‘Departments’ and optional entity ‘Absence’ and ‘Absence reasons’.

Database contains at least two (1:n)-relationships (Fig. 2) and at least one (n:m)-relationship (Fig. No. 3).


a)

SKLEPP.EMPLOYEES

P	*	ID	NUMBER
	*	FIRST_NAME	VARCHAR2 (30 BYTE)
	*	SURNAME	VARCHAR2 (60 BYTE)
		PESEL	VARCHAR2 (11 BYTE)
	*	SEX	VARCHAR2 (1 BYTE)
		BIRTH_DATE	DATE
		SALARY	NUMBER
F		DEPARTMENT_ID	NUMBER

PRACOWNICY_PK (ID)

SYS_C009475 (DEPARTMENT_ID)

PRACOWNICY_PK (ID)

SKLEPP.DEPARTMENTS

P	*	DEPARTMENT_CODE	NUMBER
		DEPARTMENT_NAME	VARCHAR2 (40 BYTE)
F	*	ID_KIEROWNIKA	NUMBER
		CITY	VARCHAR2 (50 BYTE)

DEPARTAMENTY_PK (DEPARTMENT_CODE)

FK_DEP_PRACOWNIK (ID_KIEROWNIKA)

DEPARTAMENTY_PK (DEPARTMENT_CODE)

b)

SKLEPP.SUPPLIERES

P	*	SUPPLIER_ID	NUMBER
	*	SUPPLIER_NAME	VARCHAR2 (50 BYTE)
F		CODE	VARCHAR2 (10 BYTE)
		STATUS	VARCHAR2 (1 BYTE)

SUPPLIERES_PK (SUPPLIER_ID)

SYS_C009498 (CODE)

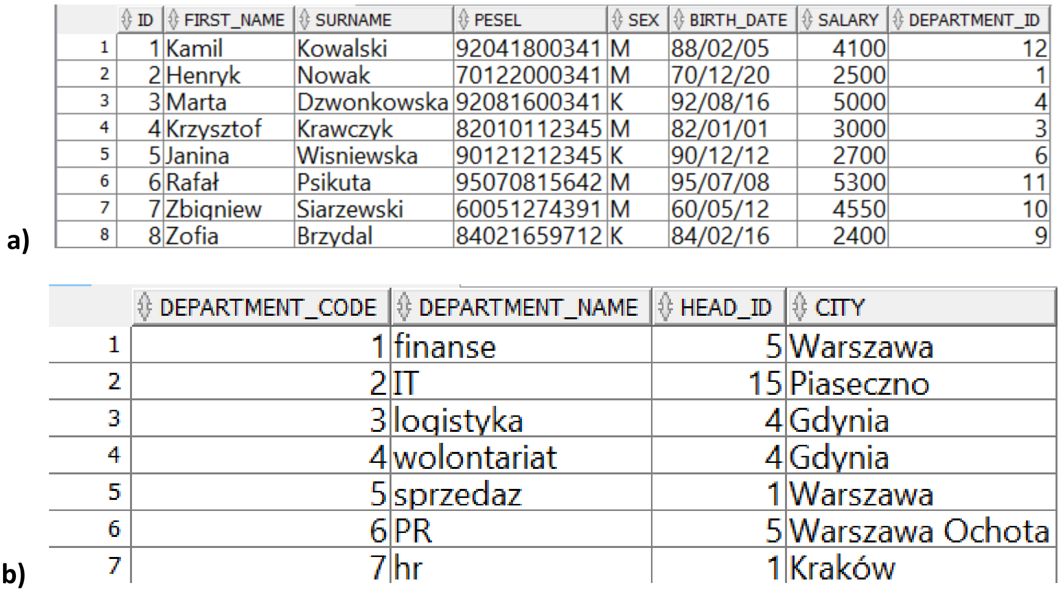
SKLEPP.MEANS_OF_TRANSPORT

P	*	CODE	VARCHAR2 (10 BYTE)
	*	FULL_NAME	VARCHAR2 (80 BYTE)
		MIN_LENGHT	NUMBER
		MAX_DLUGOSC	NUMBER
		MIN_WEIGHT	NUMBER
		MAX_WEIGHT	NUMBER
		CAPACITY	NUMBER

MEANS_OF_TRANSPORT_PK (CODE)

Fig. 2. a) 'Employees' – 'Departments' b) 'Suppliers' – 'Means of transport' 1:n relationships.

1:n relations presented in Fig. 2. have been accomplished through Primary and Foreign Keys connecting both tables. Relationship a) connects employee to one of the departments one belongs to, whereas relationship b) allows to explore knowledge about planned transport (for example capacity of one transport).


a)

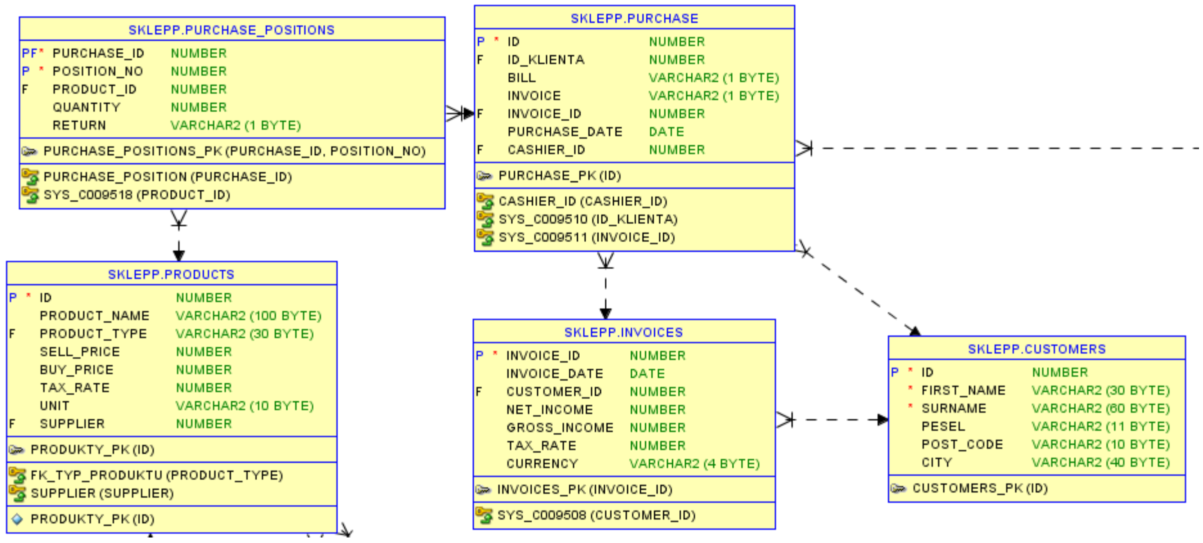
	ID	FIRST_NAME	SURNAME	PESEL	SEX	BIRTH_DATE	SALARY	DEPARTMENT_ID
1	1	Kamil	Kowalski	92041800341	M	88/02/05	4100	12
2	2	Henryk	Nowak	70122000341	M	70/12/20	2500	1
3	3	Marta	Dzwonkowska	92081600341	K	92/08/16	5000	4
4	4	Krzysztof	Krawczyk	82010112345	M	82/01/01	3000	3
5	5	Janina	Wisniewska	90121212345	K	90/12/12	2700	6
6	6	Rafał	Psikuta	95070815642	M	95/07/08	5300	11
7	7	Zbigniew	Siarzewski	60051274391	M	60/05/12	4550	10
8	8	Zofia	Brzydal	84021659712	K	84/02/16	2400	9

b)

	DEPARTMENT_CODE	DEPARTMENT_NAME	HEAD_ID	CITY
1	1	finanse	5	Warszawa
2	2	IT	15	Piaseczno
3	3	logistyka	4	Gdynia
4	4	wolontariat	4	Gdynia
5	5	sprzedaz	1	Warszawa
6	6	PR	5	Warszawa Ochota
7	7	hr	1	Kraków

Fig. 3. Sections of related tables a) 'Employees' b) 'Departments'.

Fig. 3. presents sections of both tables connected with primary keys: ID – for 'Employees' table and DEPARTMENT_CODE for 'Departments' that are ihnerited by foreign keys: HEAD_ID and DEPARTMENT_ID respectively. This composition provides double-inheritance which is assupted to be convinient in deep human resources analisys.



SKLEPP.PURCHASE_POSITIONS		
PF*	PURCHASE_ID	NUMBER
P *	POSITION_NO	NUMBER
F	PRODUCT_ID	NUMBER
	QUANTITY	NUMBER
	RETURN	VARCHAR2 (1 BYTE)
	PURCHASE_POSITIONS_PK (PURCHASE_ID, POSITION_NO)	
	PURCHASE_POSITION (PURCHASE_ID)	
	SYS_C009518 (PRODUCT_ID)	

SKLEPP.PURCHASE		
P *	ID	NUMBER
F	ID_KLIENTA	NUMBER
	BILL	VARCHAR2 (1 BYTE)
	INVOICE	VARCHAR2 (1 BYTE)
F	INVOICE_ID	NUMBER
	PURCHASE_DATE	DATE
F	CASHIER_ID	NUMBER
	PURCHASE_PK (ID)	
	CASHIER_ID (CASHIER_ID)	
	SYS_C009510 (ID_KLIENTA)	
	SYS_C009511 (INVOICE_ID)	

SKLEPP.PRODUCTS		
P *	ID	NUMBER
	PRODUCT_NAME	VARCHAR2 (100 BYTE)
F	PRODUCT_TYPE	VARCHAR2 (30 BYTE)
	SELL_PRICE	NUMBER
	BUY_PRICE	NUMBER
	TAX_RATE	NUMBER
	UNIT	VARCHAR2 (10 BYTE)
F	SUPPLIER	NUMBER
	PRODUKTY_PK (ID)	
	FK_TYP_PRODUKTU (PRODUCT_TYPE)	
	SUPPLIER (SUPPLIER)	
	PRODUKTY_PK (ID)	

SKLEPP.INVOICES		
P *	INVOICE_ID	NUMBER
	INVOICE_DATE	DATE
F	CUSTOMER_ID	NUMBER
	NET_INCOME	NUMBER
	GROSS_INCOME	NUMBER
	TAX_RATE	NUMBER
	CURRENCY	VARCHAR2 (4 BYTE)
	INVOICES_PK (INVOICE_ID)	
	SYS_C009508 (CUSTOMER_ID)	

SKLEPP.CUSTOMERS		
P *	ID	NUMBER
*	FIRST_NAME	VARCHAR2 (30 BYTE)
*	SURNAME	VARCHAR2 (60 BYTE)
	PESEL	VARCHAR2 (11 BYTE)
	POST_CODE	VARCHAR2 (10 BYTE)
	CITY	VARCHAR2 (40 BYTE)
	CUSTOMERS_PK (ID)	

Fig. 4. 'Products'-'Customers' n:m relationship.

Database posses many-to-many relationship and one of them was presented in Fig. 4. Each row in table 'Customers' can reference to many rows in table 'Products', and each row in table 'Products' can reference to many rows in table 'Customers'. These inheritances can be easily imagined by any customer purchasing several products, but one product may be sold to different customers. Technically it is realized by sequence of entities with primary keys referencing one to another. The most significant points on the road from both tables are primary keys that foreign keys are reffered to: CUSTOMER_ID ('CUSTOMERS') referred to <- PURCHASE_ID ('PURCHASE') reffered to <- PURCHASE_ID (storing list of purchased products on one bill in 'PURCHASE_POSITIONS') with refferation to PRODUCT_ID in 'PRODUCTS'.

The structure of the database has been created through writing a code, with commands as follows:

- **Table creation:**

```
CREATE TABLE invoices (invoice_id NUMBER,
                       ivoice_date DATE,
                       customer_id NUMBER,
                       net_income NUMBER,
                       gross_income NUMBER,
                       tax_rate NUMBER,
                       currency VARCHAR2(4));
```

- **Data inserting:**

```
INSERT INTO invoices (invoice_id,
                      invoice_date,
                      customer_id,
                      net_income,
                      gross_income,
                      tax_rate,
                      currency)
VALUES (2,
        '22/03/20',
        4,
        10,
        10.23,
        23,
        'PLN');
```

- **Modifing constraints:**

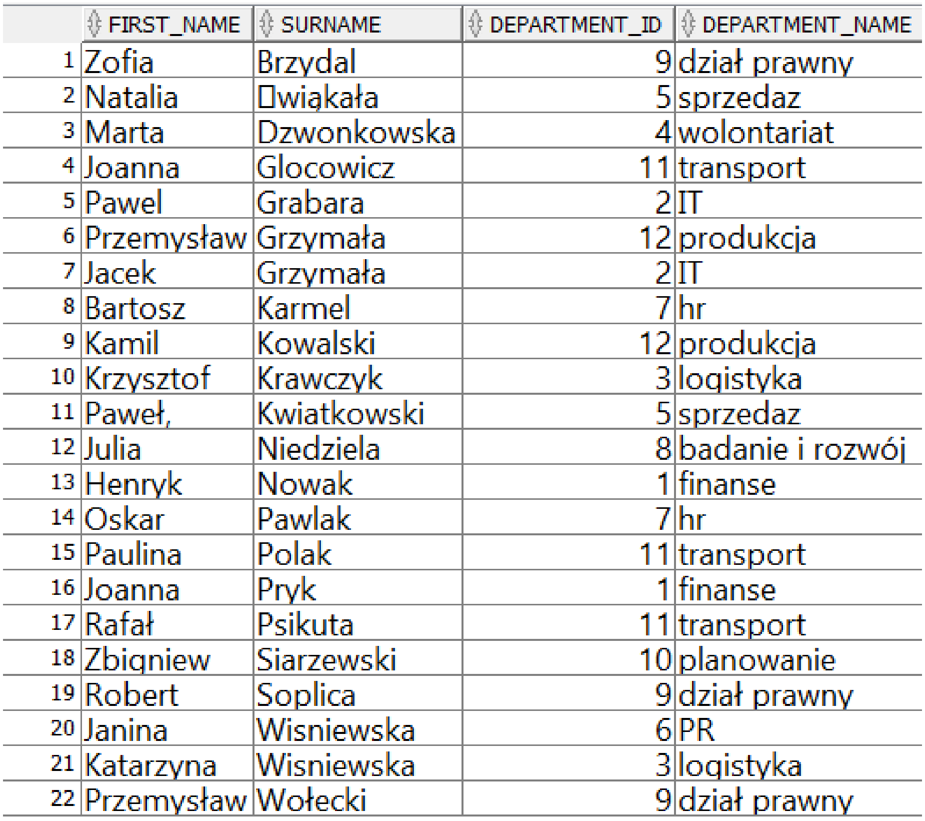
```
ALTER TABLE invoices MODIFY net_income NOT NULL;
```

The example DDL code for ‘Employees’ entity has been pasted below:

```
--------------------------------------------------------
--  DDL for Table EMPLOYEES
--------------------------------------------------------

  CREATE TABLE "SKLEPP"."EMPLOYEES"
   (        "ID" NUMBER,
            "FIRST_NAME" VARCHAR2(30 BYTE),
            "SURNAME" VARCHAR2(60 BYTE),
            "PESEL" VARCHAR2(11 BYTE),
            "SEX" VARCHAR2(1 BYTE),
            "BIRTH_DATE" DATE,
            "SALARY" NUMBER,
            "DEPARTMENT_ID" NUMBER
   ) SEGMENT CREATION IMMEDIATE
  PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
 NOCOMPRESS LOGGING
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  DDL for Index PRACOWNICY_PK
--------------------------------------------------------

  CREATE UNIQUE INDEX "SKLEPP"."PRACOWNICY_PK" ON "SKLEPP"."EMPLOYEES" ("ID")
  PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS" ;
--------------------------------------------------------
--  Constraints for Table EMPLOYEES
--------------------------------------------------------

  ALTER TABLE "SKLEPP"."EMPLOYEES" MODIFY ("FIRST_NAME" NOT NULL ENABLE);
  ALTER TABLE "SKLEPP"."EMPLOYEES" MODIFY ("SURNAME" NOT NULL ENABLE);
  ALTER TABLE "SKLEPP"."EMPLOYEES" ADD CHECK ("SEX"='M' OR "SEX"='K') ENABLE;
  ALTER TABLE "SKLEPP"."EMPLOYEES" MODIFY ("SEX" NOT NULL ENABLE);
  ALTER TABLE "SKLEPP"."EMPLOYEES" ADD CONSTRAINT "PRACOWNICY_PK" PRIMARY KEY ("ID")
  USING INDEX PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
  PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
  BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
  TABLESPACE "USERS"  ENABLE;
--------------------------------------------------------
--  Ref Constraints for Table EMPLOYEES
--------------------------------------------------------

  ALTER TABLE "SKLEPP"."EMPLOYEES" ADD FOREIGN KEY ("DEPARTMENT_ID")
	  REFERENCES "SKLEPP"."DEPARTMENTS" ("DEPARTMENT_CODE") ENABLE;
```

Data statistics can be described by simple JOIN command allowing to present referrences between tables that connection is easily noticeable, following query presents outcome that is presented in Fig. 5:

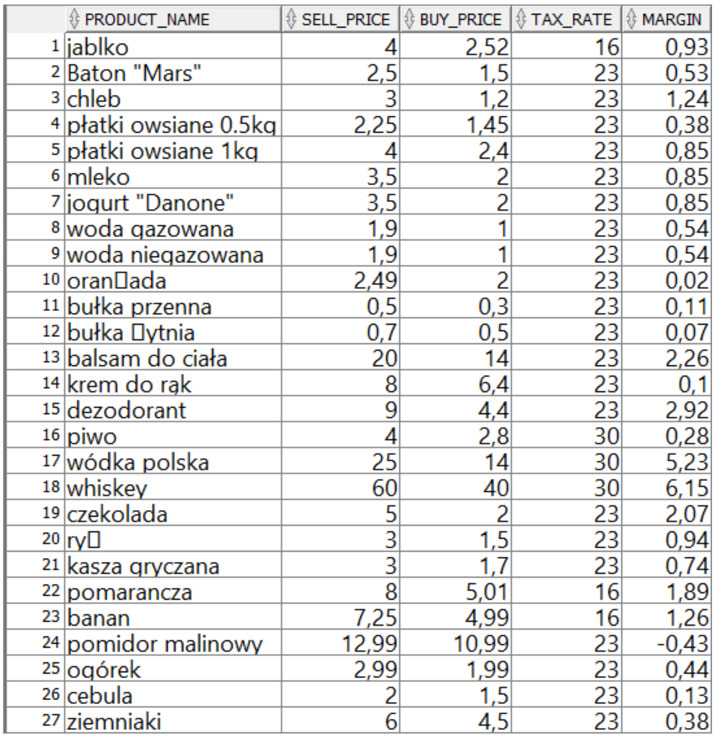
```
SELECT first_name, surname, department_id, department_name

FROM employees

INNER JOIN departments

ON employees.department_id = departments.department_code

ORDER BY surname;
```



	FIRST_NAME	SURNAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Zofia	Brzydal	9	dział prawny
2	Natalia	□wiąkała	5	sprzedaz
3	Marta	Dzwonkowska	4	wolontariat
4	Joanna	Glocowicz	11	transport
5	Pawel	Grabara	2	IT
6	Przemysław	Grzymała	12	produkcja
7	Jacek	Grzymała	2	IT
8	Bartosz	Karmel	7	hr
9	Kamil	Kowalski	12	produkcja
10	Krzysztof	Krawczyk	3	logistyka
11	Paweł,	Kwiatkowski	5	sprzedaz
12	Julia	Niedziela	8	badanie i rozwój
13	Henryk	Nowak	1	finanse
14	Oskar	Pawlak	7	hr
15	Paulina	Polak	11	transport
16	Joanna	Pryk	1	finanse
17	Rafał	Psikuta	11	transport
18	Zbigniew	Siarzewski	10	planowanie
19	Robert	Soplica	9	dział prawny
20	Janina	Wisniewska	6	PR
21	Katarzyna	Wisniewska	3	logistyka
22	Przemysław	Wołecki	9	dział prawny

Fig. 5. Result of query including join command.

Quantative analysis of collected data can bring conclusions on detailed statistics such as margin of each product stored. This has been shown in Fig. 6.



	PRODUCT_NAME	SELL_PRICE	BUY_PRICE	TAX_RATE	MARGIN
1	jablko	4	2,52	16	0,93
2	Baton "Mars"	2,5	1,5	23	0,53
3	chleb	3	1,2	23	1,24
4	płatki owsiane 0.5kg	2,25	1,45	23	0,38
5	płatki owsiane 1kg	4	2,4	23	0,85
6	mleko	3,5	2	23	0,85
7	jogurt "Danone"	3,5	2	23	0,85
8	woda gazowana	1,9	1	23	0,54
9	woda niegazowana	1,9	1	23	0,54
10	oran□ada	2,49	2	23	0,02
11	bułka przenna	0,5	0,3	23	0,11
12	bułka □ytnia	0,7	0,5	23	0,07
13	balsam do ciała	20	14	23	2,26
14	krem do rąk	8	6,4	23	0,1
15	dezodorant	9	4,4	23	2,92
16	piwo	4	2,8	30	0,28
17	wódka polska	25	14	30	5,23
18	whiskey	60	40	30	6,15
19	czekolada	5	2	23	2,07
20	ry□	3	1,5	23	0,94
21	kasza gryczana	3	1,7	23	0,74
22	pomarancza	8	5,01	16	1,89
23	banan	7,25	4,99	16	1,26
24	pomidor malinowy	12,99	10,99	23	-0,43
25	ogórek	2,99	1,99	23	0,44
26	cebula	2	1,5	23	0,13
27	ziemniaki	6	4,5	23	0,38

Fig. 6. Example statistics of the data.

Data presented in Fig. 6. was generated by following query:

```
SELECT product_name,
       sell_price,
       buy_price,
       tax_rate,
       round((100*sell_price/((100+tax_rate))-buy_price), 2) margin
       FROM products
```