

JEDNODUCHÁ DB: SVETOVE_OBJEDNAVKY

E : ZAKAZNICI (**ID_ZAK**, JMENO, PRIJMENI, MESTO, ZEME, TELEFON) 3NF

E : OBJEDNAVKY (**ID_OBJ**, DATUM_OBJ, CISLO_OBJ, CELKOVA_CENA, **ZAKAZNIK(FK)**) 3NF

E : PRODEJCI (**ID_PCI**, SPOLECNOST, KONTAKT_OS, MESTO, ZEME, TELEFON) 3NF

E : PRODUKTY (**ID_PRO**, NAZEV, POPIS, CENA, **PRODEJCE(FK)**) 3NF

R : OBJEDNAL (ZAKAZNICI, OBJEDNAVKY) 1:N

1 → N
1 ← 1

R : MA (PRODEJCI, PRODUKTY) 1:N

1 → N
1 ← 1

R : OBSAHUJI (OBJEDNAVKY, PRODUKTY) M:N

1 → N
N ← 1

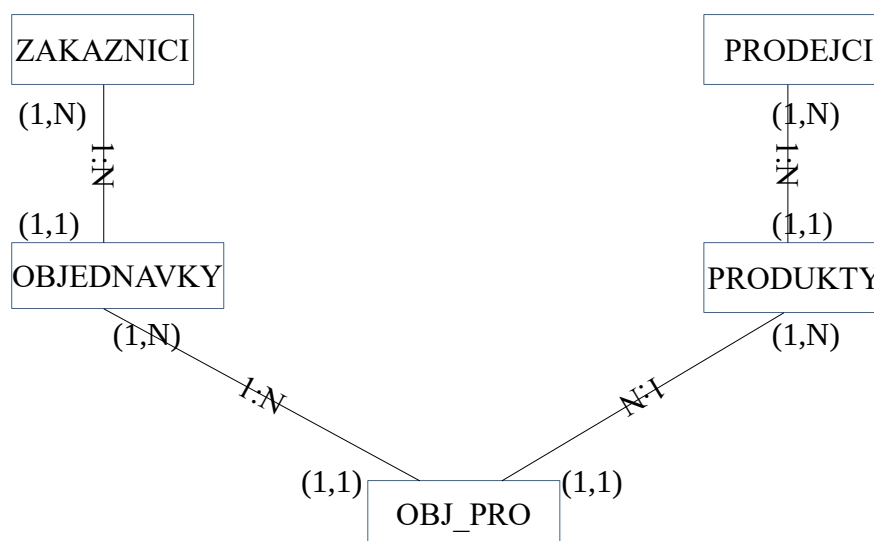
E : OBJ_PRO (**OBJEDNAVKA(FK)**, **PRODUKT(FK)**, **POCET_KUSU**, **ID_O_P**) 3NF

CENA > 0

DATUM_OBJ ≤ dnesni_datum

|TELEFON| = 9 cifer

POCET_KUSU ≥ 1



E: ZAKAZNICI

ID_ZAK	JMENO	PRIJMENI	MESTO	ZEME	TEEFON
1	Milan	Mroczkowski	Žatec	Česká republika	776555405
2	Pavel	Doubrava	Kadaň	Česká republika	776555444
3	Aleš	Silný	Most	Česká republika	776777555
4	Paul	Braun	Hamburg	Deutschland	605333402

E: OBJEDNAVKY

ID_OBJ	DATUM_OBJ	CISLO_OBJ	CELKOVA_CENA	ZAKAZNIK
1	2014-05-22	7248	2500	1
2	2012-06-13	7695	2000	2
3	2013-07-26	1645	3000	3
4	2014-05-22	8753	500	4

E: PRODEJCI

ID_PCI	SPOLECNOST	KONTAKT_OS	MESTO	ZEME	TELEFON
1	CZC	Ivan Novák	Kadaň	Česká republika	776555421
2	ALZA	Ivana Stará	Chomutov	Česká republika	776322422
3	TS	Ind Slav	Most	Česká republika	776111223
4	AIR	Otto Malbeck	Hamburg	Deutschland	723777666
5	MOBIL	Silva Svarc		Frankreich	732111229
6	MOBIL	Patrik Dlouhý	Most	Česká republika	732999111

E: PRODUKTY

ID_PRO	NAZEV	POPIS	CENA	PRODEJCE
1	LCD TX	Samsung	500	1
2	LCD SVC	Sony	500	2
3	OLED	Samsung	1000	3
4	LCD LV	TLC	100	4

E: OBJ_PRO

OBJEDNAVKA	PRODUKT	POCET_KUSU	ID_O_P
1	1	5	1
2	2	4	2
3	3	3	3
4	4	5	4

Pozn.: data jsou vymyšlená.

```
CREATE DATABASE svetove_objednavky;
USE svetove_objednavky;
```

```
CREATE TABLE zakaznici (id_zak int auto_increment,
                        jmeno varchar(20) NOT NULL,
                        prijmeni varchar(20) NOT NULL,
                        mesto varchar(40),
                        zeme varchar(40),
                        telefon char(9),
                        primary key(id_zak) );
```

```
CREATE TABLE objednavky (id_obj int auto_increment,
                          datum_obj date NOT NULL,
                          cislo_obj int(11) NOT NULL,
                          celkova_cena int(11),
                          zakaznik int,
                          primary key(id_obj),
                          foreign key (zakaznik) references zakaznici(id_zak),
                          CONSTRAINT datum CHECK (datum_obj BETWEEN 0 and
(sysdate,'YYYY-MM-DD')) );
```

```
CREATE TABLE prodejci (id_pci int auto_increment,
                        spolecnost varchar(20) NOT NULL,
                        kontakt_os varchar(20) NOT NULL,
                        mesto varchar(40),
                        zeme varchar(40),
                        telefon char(9),
                        primary key(id_pci) );
```

```
CREATE TABLE produkty (id_pro int auto_increment,
                        nazev varchar(20) NOT NULL,
                        popis varchar(20) NOT NULL,
                        cena int(11),
                        prodejce int,
                        primary key(id_pro),
                        foreign key(prodejce) references prodejci(id_pci) );
```

```
CREATE TABLE obj_pro (id_o_p int auto_increment,
                       objednavka int,
                       produkt int,
                       pocet_kusu int,
                       primary key(id_o_p),
                       foreign key(objednavka) references objednavky(id_obj),
                       foreign key(produkt) references produkty(id_pro) );
```

```
SHOW DATABASES;
SHOW TABLES;
DELETE FROM zakaznici;
```

```
INSERT INTO zakaznici (id_zak, jmeno, prijmeni, mesto, zeme, telefon)
VALUES (1, 'Milan', 'Mroczkowski', 'Žatec', 'Česká republika', 776555405);
INSERT INTO zakaznici (id_zak, jmeno, prijmeni, mesto, zeme, telefon)
VALUES (2, 'Pavel', 'Doubrava', 'Kadaň', 'Česká republika', 776555444);
INSERT INTO zakaznici (id_zak, jmeno, prijmeni, mesto, zeme, telefon)
VALUES (3, 'Aleš', 'Silný', 'Most', 'Česká republika', 776777555);
INSERT INTO zakaznici (id_zak, jmeno, prijmeni, mesto, zeme, telefon)
VALUES (4, 'Paul', 'Braun', 'Hamburg', 'Deutschland', 776777555);
```

```
INSERT INTO objednavky (id_obj, datum_obj, cislo_obj, celkova_cena, zakaznik)
VALUES (1, '2014-05-22', 7248, 2500, 1);
INSERT INTO objednavky (id_obj, datum_obj, cislo_obj, celkova_cena, zakaznik)
VALUES (2, '2012-06-13', 7695, 2000, 2);
INSERT INTO objednavky (id_obj, datum_obj, cislo_obj, celkova_cena, zakaznik)
VALUES (3, '2013-07-26', 1645, 3000, 3);
INSERT INTO objednavky (id_obj, datum_obj, cislo_obj, celkova_cena, zakaznik)
VALUES (4, '2014-05-22', 8753, 500, 4);
```

```
INSERT INTO prodejci (id_pci, spolecnost, kontakt_os, mesto, zeme, telefon)
VALUES (1, 'CZC', 'Ivan Novák', 'Kadaň', 'Česká republika', 776555421);
INSERT INTO prodejci (id_pci, spolecnost, kontakt_os, mesto, zeme, telefon)
VALUES (2, 'ALZA', 'Ivana Stará', 'Chomutov', 'Česká republika', 776322422);
INSERT INTO prodejci (id_pci, spolecnost, kontakt_os, mesto, zeme, telefon)
VALUES (3, 'TS', 'Ind Slav', 'Most', 'Česká republika', 776111223);
INSERT INTO prodejci (id_pci, spolecnost, kontakt_os, mesto, zeme, telefon)
VALUES (4, 'AIR', 'Otto Malbeck', 'Hamburg', 'Deutschland', 723777666);
INSERT INTO prodejci (id_pci, spolecnost, kontakt_os, mesto, zeme, telefon)
VALUES (5, 'MOBIL', 'Silva Svarc', '', 'Frankreich', 732111229);
INSERT INTO prodejci (id_pci, spolecnost, kontakt_os, mesto, zeme, telefon)
VALUES (6, 'MOBIL', 'Patrik Dlouhý', 'Most', 'Česká republika', 732999111);
```

```
INSERT INTO produkty (id_pro, nazev, popis, cena, prodejce)
VALUES (1, 'LCD TX', 'Samsung', '500', 1);
INSERT INTO produkty (id_pro, nazev, popis, cena, prodejce)
VALUES (2, 'LCD SVC', 'Sony', '500', 2);
INSERT INTO produkty (id_pro, nazev, popis, cena, prodejce)
VALUES (3, 'OLED', 'Samsung', '1000', 3);
INSERT INTO produkty (id_pro, nazev, popis, cena, prodejce)
VALUES (4, 'LCD LV', 'TLC', '100', 4);
```

```
INSERT INTO obj_pro (objednavka, produkt, pocet_kusu, id_o_p)
VALUES (1, 1, 5, 1);
INSERT INTO obj_pro (objednavka, produkt, pocet_kusu, id_o_p)
VALUES (2, 2, 4, 2);
INSERT INTO obj_pro (objednavka, produkt, pocet_kusu, id_o_p)
VALUES (3, 3, 3, 3);
INSERT INTO obj_pro (objednavka, produkt, pocet_kusu, id_o_p)
VALUES (4, 4, 5, 4);
```

Poddotazy IN

```
SELECT id_pci, spolecnost, kontakt_os, mesto, zeme, telefon  
FROM prodejci  
WHERE mesto IN (SELECT mesto FROM zakaznici);
```

Pokud se nějaký prodejce nachází ve městě ve kterém žádný zákazník neexistuje, tak se záznam prodejce nevypíše. Tedy, pomocí IN mohu provést určitou kontrolu. Žádný zákazník není v Chomutově, u jednoho prodavače nevím z jakého města je.

ID_PCI	SPOLECNOST	KONTAKT_OS	MESTO	ZEME	TELEFON
1	CZC	Ivan Novák	Kadaň	Česká republika	776555421
3	TS	Ind Slav	Most	Česká republika	776111223
4	AIR	Otto Malbeck	Hamburg	Deutschland	723777666
6	MOBIL	Patrik Dlouhý	Most	Česká republika	732999111

```
SELECT id_pci, spolecnost, kontakt_os, mesto, zeme, telefon  
FROM prodejci  
WHERE mesto IN ('Most', 'Hamburg');
```

ID_PCI	SPOLECNOST	KONTAKT_OS	MESTO	ZEME	TELEFON
3	TS	Ind Slav	Most	Česká republika	776111223
4	AIR	Otto Malbeck	Hamburg	Deutschland	723777666
6	MOBIL	Patrik Dlouhý	Most	Česká republika	732999111

```
SELECT id_pci, spolecnost, kontakt_os, mesto, zeme, telefon  
FROM prodejci  
WHERE mesto NOT IN ('Most', 'Hamburg');
```

ID_PCI	SPOLECNOST	KONTAKT_OS	MESTO	ZEME	TELEFON
1	CZC	Ivan Novák	Kadaň	Česká republika	776555421
2	ALZA	Ivana Stará	Chomutov	Česká republika	776322422
5	MOBIL	Silva Svarc		Frankreich	732111229

Poddotazy EXISTS

```
SELECT id_pci, spolecnost, kontakt_os, mesto, zeme, telefon
FROM prodejci
WHERE NOT EXISTS (SELECT mesto FROM zakaznici WHERE prodejci.mesto=
zakaznici.mesto);
```

ID_PCI	SPOLECNOST	KONTAKT_OS	MESTO	ZEME	TELEFON
2	ALZA	Ivana Stará	Chomutov	Česká republika	776322422
5	MOBIL	Silva Svarc		Frankreich	732111229

Vybral jsem vše z tabulky prodejci, mohl jsem místo vypisování atributů použít *, ale vypíše pouze záznamy z tabulky prodejci kde neexistuje shodné město v tabulce zakaznici.

```
SELECT id_zak, jmeno, prijmeni, mesto, zeme
FROM zakaznici
WHERE NOT EXISTS (SELECT mesto FROM prodejci WHERE zakaznici.mesto=
prodejci.mesto);
```

ID_ZAK	JMENO	PRIJMENI	MESTO	ZEME	TEEFON
1	Milan	Mroczkowski	Žatec	Česká republika	776555405

Vybral jsem vše z tabulky zakaznici, mohl jsem místo vypisování atributů použít *, ale vypíše pouze záznamy z tabulky zakaznici kde neexistuje shodné město v tabulce prodejci.

Pokud by bylo pouze exists, tedy bez not, tak by vypsalo to co nevypsalo a to co vypsalo by nevypsalo.

Poddotazy ANY (SOME má stejný význam)

```
SELECT id_pci, spolecnost, kontakt_os, mesto, zeme, telefon  
FROM prodejci  
WHERE mesto = ANY (SELECT mesto FROM zakaznici);
```

ID_PCI	SPOLECNOST	KONTAKT_OS	MESTO	ZEME	TELEFON
1	CZC	Ivan Novák	Kadaň	Česká republika	776555421
3	TS	Ind Slav	Most	Česká republika	776111223
4	AIR	Otto Malbeck	Hamburg	Deutschland	723777666
6	MOBIL	Patrik Dlouhý	Most	Česká republika	732999111

"= ANY" je ekvivalent k IN

ANY používáme hlavně k porovnávání číselných hodnot.

Poddotazy ANY a ALL

Vytvořím si dvě třídy entit (pro ukázkou), kde provedu porovnání pomocí dotazů s ALL:

```
CREATE TABLE technici (id_tech int auto_increment,  
                        jmeno varchar(20) NOT NULL,  
                        prijmeni varchar(20) NOT NULL,  
                        plat int,  
                        zeme varchar(40),  
                        primary key(id_tech) );
```

```
CREATE TABLE ostatni_zam (id_zam int auto_increment,  
                           jmeno varchar(20) NOT NULL,  
                           prijmeni varchar(20) NOT NULL,  
                           plat int,  
                           zeme varchar(40),  
                           profese varchar(40),  
                           primary key(id_zam) );
```

```
INSERT INTO technici (id_tech, jmeno, prijmeni, plat, zeme)  
VALUES (1, 'Milan', 'Mroczkowski', 25000, 'Česká republika');  
INSERT INTO technici (id_tech, jmeno, prijmeni, plat, zeme)  
VALUES (2, 'Pavel', 'Doubrava', 21000, 'Česká republika');  
INSERT INTO technici (id_tech, jmeno, prijmeni, plat, zeme)  
VALUES (3, 'Aleš', 'Silný', 23000, 'Česká republika');  
INSERT INTO technici (id_tech, jmeno, prijmeni, plat, zeme)  
VALUES (4, 'Paul', 'Braun', 35000, 'Deutschland');
```

```
INSERT INTO ostatni_zam (id_zam, jmeno, prijmeni, plat, zeme, profese)  
VALUES (1, 'Igor', 'Nový', 24000, 'Česká republika', 'opravář');  
INSERT INTO ostatni_zam (id_zam, jmeno, prijmeni, plat, zeme, profese)  
VALUES (2, 'Iveta', 'Kotoučová', 25000, 'Česká republika', 'sekretářka');  
INSERT INTO ostatni_zam (id_zam, jmeno, prijmeni, plat, zeme, profese)  
VALUES (3, 'Simona', 'Koutná', 36000, 'Česká republika', 'právník');  
INSERT INTO ostatni_zam (id_zam, jmeno, prijmeni, plat, zeme, profese)  
VALUES (4, 'Paul', 'Braun', 21000, 'Deutschland', 'účetní');  
INSERT INTO ostatni_zam (id_zam, jmeno, prijmeni, plat, zeme, profese)  
VALUES (5, 'Jan', 'Svatý', 40000, 'Česká republika', 'ředitel');
```


E: TECHNICI

id_tech	jmeno	prijmeni	plat	zeme
1	Milan	Mroczkowski	25000	Česká republika
2	Pavel	Doubrava	21000	Česká republika
3	Aleš	Silný	23000	Česká republika
4	Paul	Braun	35000	Deutschland

E: OSTATNI_ZAM

id_zam	jmeno	prijmeni	plat	zeme	profese
1	Igor	Nový	24000	Česká republika	opravář
2	Iveta	Kotoučová	25000	Česká republika	sekretářka
3	Simona	Koutná	36000	Česká republika	právník
4	Paul	Braun	21000	Deutschland	účetní
5	Jan	Svatý	40000	Česká republika	ředitel

```
SELECT id_zam, jmeno, prijmeni, plat, zeme, profese
FROM ostatni_zam
WHERE plat > ANY (SELECT plat FROM technici);
```

id_zam	jmeno	prijmeni	plat	zeme	profese
1	Igor	Nový	24000	Česká republika	opravář
2	Iveta	Kotoučová	25000	Česká republika	sekretářka
3	Simona	Koutná	36000	Česká republika	právník
5	Jan	Svatý	40000	Česká republika	ředitel

> ANY () - více než minimum, vypíše všechny platy z tabulky ostatni_zam větší než nejmenší plat z tabulky technici.

```
SELECT id_zam, jmeno, prijmeni, plat, zeme, profese
FROM ostatni_zam
WHERE plat < ANY (SELECT plat FROM technici);
```

id_zam	jmeno	prijmeni	plat	zeme	profese
1	Igor	Nový	24000	Česká republika	opravář
2	Iveta	Kotoučová	25000	Česká republika	sekretářka
4	Paul	Braun	21000	Deutschland	účetní

< ANY () - méně než maximum, vypíše všechny platy z tabulky ostatni_zam menší než maximální plat z tabulky technici.

```
SELECT id_zam, jmeno, prijmeni, plat, zeme, profese
FROM ostatni_zam
WHERE plat > ALL (SELECT plat FROM technici);
```

id_zam	jmeno	prijmeni	plat	zeme	profese
3	Simona	Koutná	36000	Česká republika	právník
5	Jan	Svatý	40000	Česká republika	ředitel

> ALL () - více než maximální, vypíše všechny platy z tabulky ostatni_zam větší než maximální plat z tabulky technici.

```
SELECT id_zam, jmeno, prijmeni, plat, zeme, profese
FROM ostatni_zam
WHERE plat < ALL (SELECT plat FROM technici);
```

Takový plat v třídě entit ostatni_zam nemám.

< ALL () - méně než minimum, vypíše všechny platy z tabulky ostatni_zam menší než minimální plat z tabulky technici.

Příklad použití ANY a ALL v rámci jedné tabulky

```
CREATE TABLE platy (id_plat int auto_increment,  
                    jmeno varchar(20) NOT NULL,  
                    prijmeni varchar(20) NOT NULL,  
                    plat int,  
                    zeme varchar(40),  
                    profese varchar(40),  
                    primary key(id_plat) );  
  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (1, 'Igor', 'Nový', 24000, 'Česká republika', 'opravář');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (2, 'Iveta', 'Kotoučová', 25000, 'Česká republika', 'sekretářka');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (3, 'Pavel', 'Doubrava', 21000, 'Česká republika', 'technik');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (4, 'Simona', 'Koutná', 36000, 'Česká republika', 'právník');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (5, 'Paul', 'Braun', 21000, 'Deutschland', 'účetní');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (6, 'Jan', 'Svatý', 40000, 'Česká republika', 'ředitel');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (7, 'Pavel', 'Brad', 25000, 'Deutschland', 'technik');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (8, 'Adam', 'Spadl', 35000, 'Česká republika', 'technik');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (9, 'Aleš', 'Silný', 23000, 'Česká republika', 'technik');  
INSERT INTO platy (id_plat, jmeno, prijmeni, plat, zeme, profese)  
VALUES (10, 'Iveta', 'Smutná', 18000, 'Česká republika', 'sekretářka');
```

```
SELECT id_plat, jmeno, prijmeni, plat, zeme, profese  
FROM platy  
WHERE plat > ANY (SELECT plat FROM platy WHERE profese='technik');
```

Vypíšíou se všechny platy větší než nejmenší plat technika, který je ve výši 21 000Kč.

```
SELECT id_plat, jmeno, prijmeni, plat, zeme, profese  
FROM platy  
WHERE plat < ANY (SELECT plat FROM platy WHERE profese='technik');
```

Vypíšíou se všechny platy menší než největší plat technika, který je ve výši 35 000Kč.

```
SELECT id_plat, jmeno, prijmeni, plat, zeme, profese
FROM platy
WHERE plat > ALL (SELECT plat FROM platy WHERE profese='technik');
```

Vypíšíou se všechny platy větší než největší plat technika, který je ve výši 35 000Kč.

id_plat	jmeno	prijmeni	plat	zeme	profese
4	Simona	Koutná	36000	Česká republika	právník
6	Jan	Svatý	40000	Česká republika	ředitel

```
SELECT id_plat, jmeno, prijmeni, plat, zeme, profese
FROM platy
WHERE plat < ALL (SELECT plat FROM platy WHERE profese='technik');
```

Vypíšíou se všechny platy nižší než nejvyšší plat technika, který je ve výši 21 000Kč.

id_plat	jmeno	prijmeni	plat	zeme	profese
10	Iveta	Smutná	18000	Česká republika	sekretářka

Další dotazy:

```
SELECT id_plat, jmeno, prijmeni, plat, zeme, profese
FROM platy
WHERE plat IN (SELECT MIN(plat) FROM platy);
```

Zobrazí nejmenší plat.

```
SELECT id_plat, jmeno, prijmeni, plat, zeme, profese
FROM platy
WHERE plat < ANY
      (SELECT plat
       FROM platy
       WHERE profese = 'technik')
AND profese <> 'technik';
```

Vypíšíou se všechny nižší platy než největší plat technika, který je ve výši 35 000Kč a zároveň nevypíše platy techniků.

Nerovno: <> !=