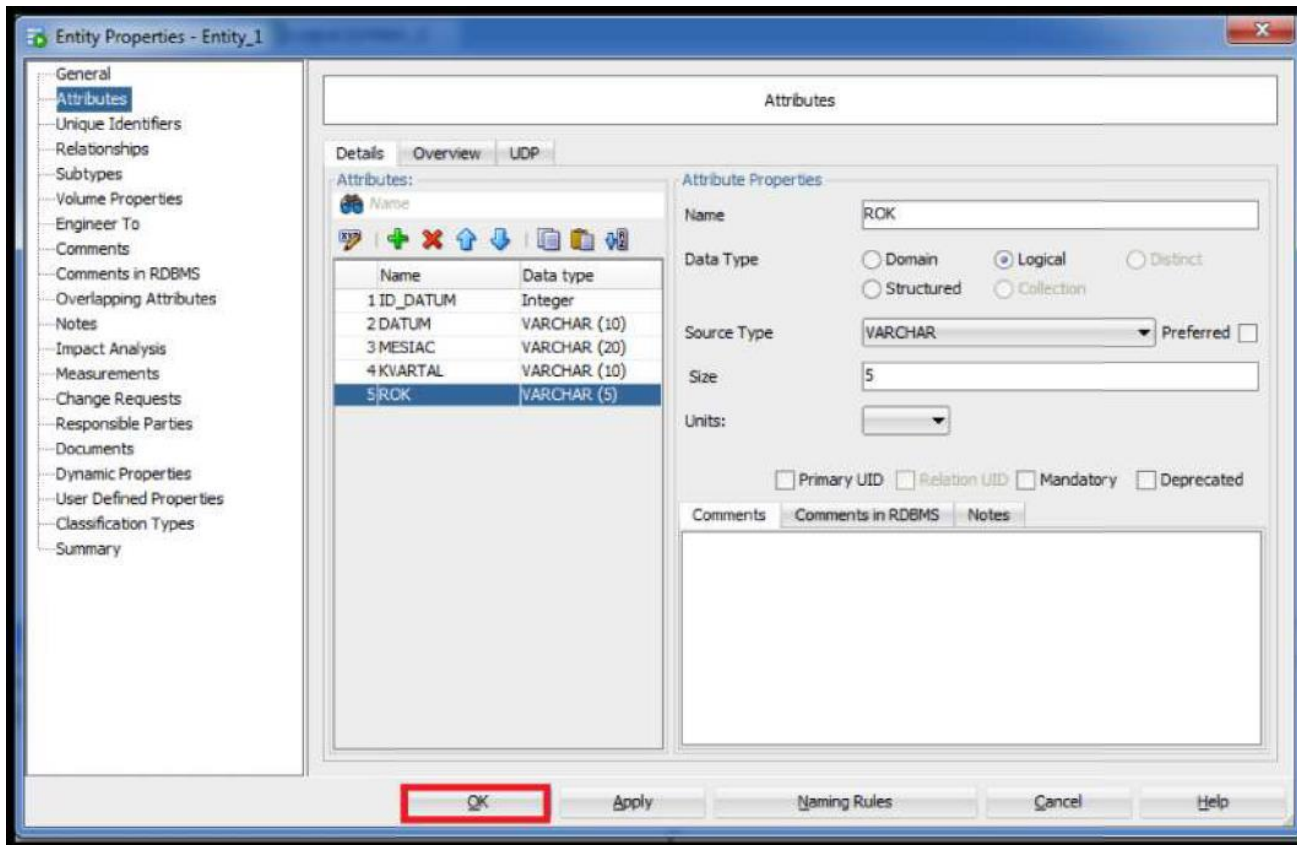


# Atribútové údaje



# Zložky geografickej informácie

- **Priestorová zložka** určuje, kde sa objekt nachádza, aký má tvar a veľkosť
- **Atribútová časť** opisuje jeho nepriestorové vlastnosti, napríklad počet obyvateľov, mená vlastníkov, maximálnu povolenú rýchlosť na úseku cesty
- Priestorová a atribútová zložka spolu tvoria geografický objekt, ktorý sa nazýva aj **entita**
- Požiadavka na zdieľaný prístup k údajom a zabezpečenie ich modifikácie

# Entita

- Je uložená ako samostatný záznam v súbore alebo v databáze
- Všetky entity rovnakého typu spolu vytvárajú **triedu objektov** a sú uložené v jednej **tematickej vrstve**
- Typ entity je charakterizovaný druhom geometrického elementu (bod, línia, polygón) a zoznamom atribútov
- K entitám je potrebné pripojiť **externé atribútové údaje**, uložené v samostatných súboroch pomocou identifikátora
- Tematické vrstvy a atribútové tabuľky spolu tvoria **geografickú databázu**

# Geografická databáza

- Je logicky usporiadaný súbor geografických údajov v digitálnej forme
- Jej obsah je daný účelom, pre ktorý bola vytvorená
- Pojem databáza sa nesmie zamieňať s termínom **báza údajov**
- Báza údajov je chápaná širšie, ako množina všetkých údajov (digitálnych aj analógových), ktoré sa používajú na riešenie

# Atribútové údaje podľa charakteru

- **Nominálne** – neusporiadaná množina hodnôt bez možnosti porovnania (mená, symboly, kategórie a pod.),
- **Ordinálne** – usporiadaná množina hodnôt s možnosťou porovnania (výška, hĺbka, šírka a pod.), pričom hodnota môže byť len klasifikovaná (malá, veľká),
- **Intervalové** – usporiadaná množina hodnôt s možnosťou porovnania pomocou súčtu alebo rozdielu (nadmorská výška, teplota, tlak a pod.), pričom množina údajov nemusí obsahovať počiatok (nemusí mať nulovú hodnotu),
- **Podielové** – usporiadaná množina hodnôt s definovaným počiatkom. Dá sa korektne vypočítať podiel dvoch údajov, napr. výmera jednej parcely je dvojnásobná oproti výmere inej parcely

# Databázové systémy

- Databázové systémy (banka údajov, databanka) sú zlúčením údajov a nástrojov pre prácu s údajmi
- Slúžia na uchovávanie, vyhľadávanie a poskytovanie údajov
- Databázový systém sa skladá zo štyroch základných častí
  - **databáza**
  - **programové vybavenie** – hlavnou časťou je systém riadenia bázy údajov. Okrem toho obsahuje programy pre tvorbu, údržbu, správu a zálohovanie databázy
  - **technické vybavenie** – zariadenia potrebné na prevádzku programového vybavenia, uchovávanie, zálohovanie a poskytovanie údajov
  - **personálne zabezpečenie a organizačné opatrenia** – osoby, ktoré zabezpečujú prevádzku a správu databázového systému podľa daných pravidiel

# Databázový model

- Databázový model sa chápe ako **určenie pravidiel**, ktoré jednoznačne definujú organizáciu údajov, štruktúru a vzťahy medzi jednotkami v databáze
- Obsah databázy definujeme pomocou typov údajov
- Na opis modelu a organizáciu údajov sa využíva databázová schéma, ktorú zapisujeme do grafických schém pomocou dohodnutých symbolov a jazykov pre definovanie údajov
- Druhy databázových modelov:
  - hierarchický
  - sieťový
  - objektovo – orientovaný
  - relačný

# Hierarchický model

- Záznamy sú prepojené v stromovej štruktúre
- Každý záznam môže mať len jeden nadradený (koreňový záznam - rodič) a niekoľko podriadených záznamov (potomok)
- Typickou stromovou štruktúrou je organizácia územných celkov:

štát – kraj – okres – obec – katastrálne územie



# Sieťový model

- Je rozšírenie hierarchického modelu pomocou uzlov a množinových štruktúr
- Na rozdiel od hierarchického modelu nie je smer a počet spojení medzi uzlami obmedzený a je umožnené aj viacnásobné spojenie
- Štruktúra záznamov vyjadruje vzťah 1:N jeden záznam (vlastník) k niekoľkým záznamom (členom). Člen však musí byť vo vzťahu k práve jednému vlastníkovi

# Objektovo - orientovaný model

- Základnou jednotkou je jeden objekt, ktorý predstavuje jednu konkrétnu entitu
- Každý objekt (inštancia) patrí do nejakej triedy objektov, ktorá opisuje vlastnosti (atribúty), ich správanie, vnútorný stav a vzťahy k iným objektom,
- Z neho vznikol neskôr **Objektovo-relačný model**

# Relačný model

- V súčasnosti najviac využívaný model
- V tomto modeli sú všetky údaje a ich vzájomné vzťahy reprezentované ako relácie (tabuľky)
- Jeden záznam zodpovedá jednej entite
- Údaje aj vzťahy medzi nimi sa ukladajú do tabuliek
- Tabuľky zodpovedajú matematickým reláciám z teórie množín, z čoho pochádza názov relačného údajového modelu
- Základnými prvkami tabuľky sú
  - hodnota
  - stĺpec
  - riadok

# Atribútová tabuľka

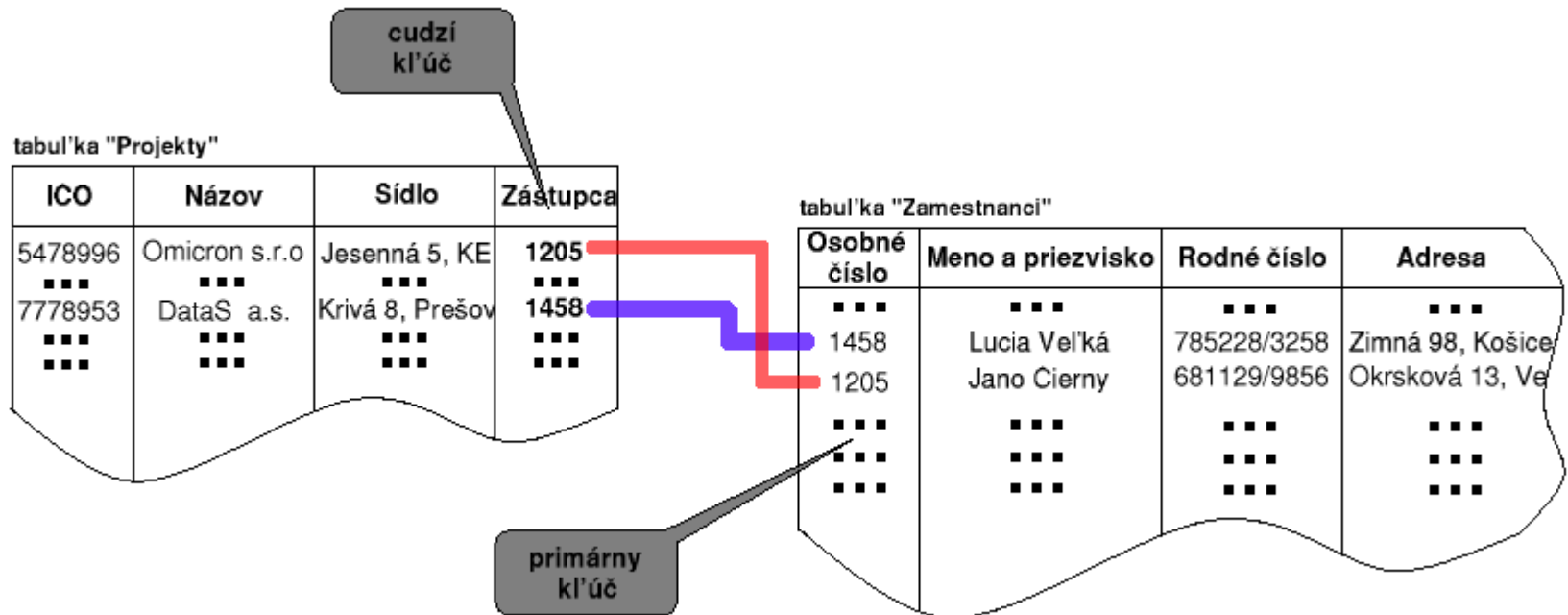
The diagram shows a table with five columns and seven rows. Callouts point to the column headers, a row, and a column.

Osobné číslo	Méno a priezvisko	Rodné číslo	Adresa	Plat
1545	Peter Malý	780513/9875	Jarná 5, Malá Čierna	7865
1458	Lucia Veľká	785228/3258	Zimná 98, Košice 8	10573
1205	Jano Čierny	681129/9856	Okrsková 13, Veľká Ida	25879
5879	Peter Okolka	510828/7149	Biele 9, Bratislava 1	9999
2001	Eva Malá	475111/3322	Uhlárska 23, Brezno	14569
...	...	...	...	...

- Stĺpce tabuľky nazývame polia (atribúty)
- Každá tabuľka musí obsahovať aspoň jedno pole
- Pole sa skladá zo začiatočných písmen alebo slabík niekoľkých slov (akronym), ktoré opisujú názov atribútu
- Pre každý atribút sa definuje údajový typ, pomocou ktorého sa vymedzuje druh a rozsah hodnôt, ktoré budú v položke uložené (text, dátum, mena, číslo, áno/nie a pod.)

- Je vhodné definovať množinu dovolených hodnôt atribútu, tzv. **doménu**
- Každý riadok databázového súboru sa nazýva záznam a obsahuje súbor hodnôt atribútov
- Pre každý atribút tabuľky musíme určiť aký **typ dát** môže obsahovať: celé čísla (numeric), znakové reťazce (charakter), logické hodnoty (logical) (áno/nie), reálne čísla, menové údaje, dátum a čas (date), ale i zložitejšie typy ako obrázky, video, audioklip, a pod.
- Báza údajov môže obsahovať väčšie množstvo tabuliek, záleží na tom, čo všetko chceme do databázy zaznamenať. Každá tabuľka má preto **svoje meno**, ktoré ju v rámci databázy jednoznačne identifikuje

- Vzťahy medzi tabuľkami sa riešia pomocou vzťahov relačnej algebry
- Odkazy medzi tabuľkami sa riešia pomocou **primárneho kľúča** – atribút, jedinečný pre každý záznam (osobné číslo, rodné číslo, a pod. ...)



# Štruktúra tabuľky

- Jednotlivé položky (atribúty) databázového súboru sú organizované v stĺpcoch
- Každý riadok súboru je súborom rôznych atribútov pre jeden objekt (v našom prípade informácie o obci) a vytvára jeden záznam databázového súboru
- Pre každý atribút je definovaný údajový typ (znakový - C, číselný - N) a veľkosť atribútu

ICZUJ	OBEC	OKRES	KRAJ	POCET_OBYV	ROZLOHA	PRVA_ZMIEN
517402	Žilina	Žilina	Žilinský	87000	8003	1208
517429	Belá	Žilina	Žilinský	3300	4620	1378
517470	Čičmany	Žilina	Žilinský	188	2561	1200
517496	Dívinka	Žilina	Žilinský	880	510	1393
517542	Fačkov	Žilina	Žilinský	705	3751	1351
517551	Gbefany	Žilina	Žilinský	1230	713	1362
517593	Horný Hričov	Žilina	Žilinský	778	545	1208
517658	Kamenná Poruba	Žilina	Žilinský	1815	1418	1368
517712	Krasňany	Žilina	Žilinský	1217	1550	1354
517739	Lietava	Žilina	Žilinský	1438	1000	1318
517763	Lutiše	Žilina	Žilinský	818	2008	1662
517917	Rajec	Žilina	Žilinský	6080	3139	1193
517925	Rajecká Lesná	Žilina	Žilinský	1279	3900	1314
517933	Rajecké Teplice	Žilina	Žilinský	2732	1185	1376
517968	Stránske	Žilina	Žilinský	720	1875	1365
517984	Strečno	Žilina	Žilinský	2640	1318	1321
518000	Svederník	Žilina	Žilinský	1000	1156	1321
518034	Teplička nad Váhom	Žilina	Žilinský	3628	1100	1267

#	Name	Type	Size	Prec	Offset
1	ICZUJ	N	6	0	2
2	OBEC	C	30	0	8
3	OKRES	C	15	0	38
4	KRAJ	C	15	0	53
5	POCET_OBYV	N	6	0	68
6	ROZLOHA	N	5	0	74
7	PRVA_ZMIEN	N	4	0	79

# Pomenovanie tabuliek a ich atribútov

- Názov má vždy začínať **písmenom**. Číslice na začiatku názvu nie sú povolené
- V názvoch sa majú používať len znaky **bez diakritiky** a nesmú sa využívať špeciálne znaky
- Dĺžka názvu musí byť primeraná využívanému systému. Príliš dlhé názvy nemusia byť v dialógu zobrazené celé, krátke názvy nevyvolajú žiadnu asociáciu
- Pri názvoch niektorých príbuzných atribútov a tabuliek je vhodné využívať **prefix** – predponu (SUR\_X, SUR\_Y SUR\_Z), prípadne zaužívané skratky



# Trojúrovňová architektúra dátových štruktúr

- **Externá (logická) úroveň** – opisuje údaje z pohľadu individuálneho užívateľa (to, čo si každý užívateľ praje a požaduje). Súčasťou externej úrovne je aj zabezpečenie neoprávnených prístupov do rôznych častí databázového systému
- **Konceptuálna úroveň** – slúži na pochopenie danej problematiky a je odvodená z externých užívateľských pohľadov. Opisuje uloženie všetkých údajov a ich vzájomné vzťahy. Okrem opisu je úlohou konceptuálnej úrovne aj odstránenie duplicity spoločných údajov

- **Interná (fyzická) úroveň** – predstavuje najnižšiu úroveň reprezentácie databázy. Zaoberá sa fyzickou pamäťovou štruktúrou uloženia a organizácie údajov v počítačovom systéme. Využíva vlastnosti konkrétneho operačného systému. Opisuje sa pomocou internej schémy ktorá obsahuje typy uložených záznamov, definície polí, definície prístupov a pod. Užívateľ s touto schémou neprichádza do styku.

# Proces tvorby databázy

- Vytvorenie **prehľadného opisu** oblasti, ktorú je potrebné riešiť.
- Návrh **konceptuálnej schémy**, podľa požiadaviek užívateľov. V tejto časti je nutné zhromaždiť všetky informácie o údajoch, ktoré sa budú ukladať v databáze a vzťahoch medzi entitami
- Vytvorí sa **zoznam všetkých typov vyžadovaných údajov**. V tejto časti je dôležité údaje opisovať nielen na všeobecnej úrovni, ale zachytiť ich čo najviac detailne. Vždy sa začína dostupnosťou existujúcich údajov. Údaje sa rozdelia do tabuliek podľa hlavných oblastí alebo tematických celkov. Každá položka sa stáva poľom a zobrazí sa ako stĺpec tabuľky

- Pre každú tabuľku sa definuje **primárny kľúč** (jedinečný identifikátor každého riadku). Primárny kľúč musí vždy obsahovať jedinečné hodnoty. Navrhnú sa vzťahy medzi jednotlivými tabuľkami a určí sa typ relačného vzťahu
- Z konceptuálneho modelu vychádza **logická štruktúra**. Do modelu sa zapracujú pohľady a požiadavky užívateľov (charakter úloh a užívateľské prostredie) s využitím vhodného údajového modelu
- **Fyzický model databázy** sa vytvára automaticky programovým vybavením systému riadenia bázy údajov. Užívatelia a správca má len obmedzené možnosti ovplyvniť organizáciu databázových súborov
- Po vypracovaní všetkých častí návrhu databázového systému nasleduje jeho **implementácia**. Zahŕňa inštaláciu systému na konkrétnom technickom zariadení a jeho uvedenie do prevádzky.

- Súčasťou implementácie musí byť **otestovanie funkčnosti systému**
- Súčasťou implementácie je **naplnenie databázy údajmi**, niekoľkými spôsobmi:
  - naplnením prázdnej databázy a súčasným odladovaním celého databázového systému,
  - konverziou existujúcich údajov (pomocou existujúcich alebo nových konverzných programov),
  - zberom nových údajov (z dôvodov dlhšieho časového obdobia zberu vzniká nebezpečenstvo vzniku nekonzistentných údajov)
- Špecifickým problémom etapy naplňania je zabezpečenie správnosti vkladáných údajov. Z tohto dôvodu musia vkladané údaje podrobovať **kontrolám**, ktoré odhalia formálne a logické chyby

- Po naplnení databázy je možné pristúpiť k testovaniu **funkčnosti databázy**. Najskôr sa overuje chod databázového systému len pre základné potreby užívateľov a postupne sa overenie rozširuje na všetky funkcie celej databázy. Základným kritériom funkčnosti je spokojnosť koncových užívateľov. Počas prevádzky databázového systému sa sleduje jeho výkon, doba odozvy, výskyt chýb, požiadavky na rozšírenie funkcií systému

# Jazyk SQL

- Jazyk SQL je univerzálnym nástrojom na prácu s relačnou databázou.
- Dnes je implementovaný vo všetkých relačných systémoch riadenia bázy údajov.
- Dovoľuje v databáze vykonávať všetky druhy operácií od vytvorenia novej tabuľky, pridávanie záznamov, aktualizáciu údajov, vyhľadávanie údajov až po riadenie transakcií, definovanie užívateľov a nastavenie prístupových práv
- Konceptia relačného databázového modelu bola vytvorená na začiatku sedemdesiatych rokov minulého storočia, používal jazyk SQL (spoločnosť IBM). Na konci sedemdesiatych rokov vznikol jeden z najúspešnejších relačných systémov riadenia bázy údajov - **Oracle**. V priebehu osemdesiatych rokov vznikali ďalšie modely s podporou jazyka SQL. Jazyk SQL bol štandardizovaný v roku 1989. Posledná aktualizovaná verzia má označenie ISO/IEC 9075:2011