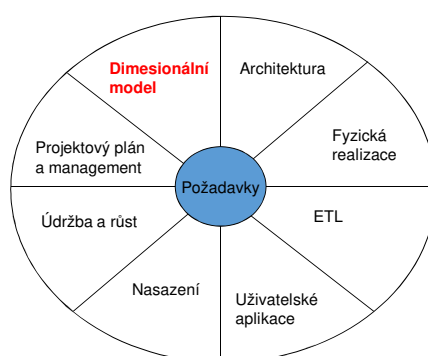


Databázové systémy a metody zpracování dat

4.přednáška

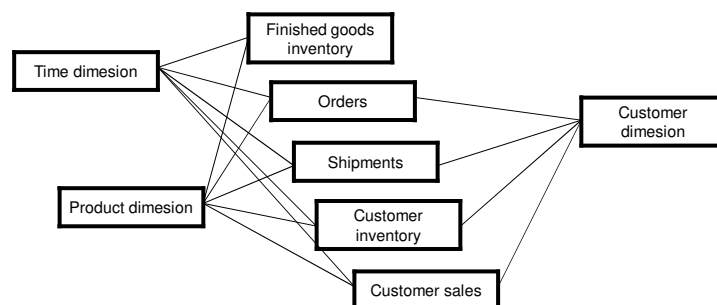


Pojmy Drill-down,

- Drill-down: ukaž mi větší detail
 - Přidání sloupce z dimenze do výstupu
- Drill-up: ukaž mi agregaci
 - Odebrání sloupce z výstupu
- Drill-across: spojení dvou a více faktových tabulek se stejnou granularitou
- Drill-around: podobné jako drill-across, ale pro nelineární uspořádání
- Slice-dice: řez multidimenzionální kostkou, omezení výběru
 - Slice – výběr dimenze (zákazník, produkt, čas)
 - Dice – výběr hodnoty v dimenzi (za rok = 2004 a produkt = chleba)

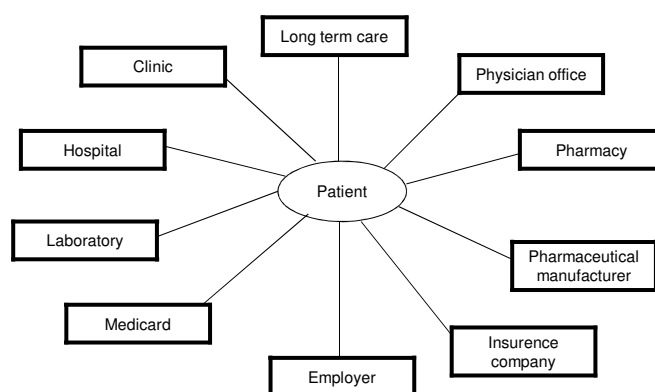
Pojmy Drill-down,

- Příklad drill-across: obchodní řetězec
 - Zásoba, objednávka, dodávka, prodej
 - Samostatné fact tabulky spojené časovou a produktovou dimenzí (dimenze jsou „conformed“ vůči faktovým tabulkám)
 - Výsledný dotaz využívá outer-join



Pojmy Drill-down,

- Drill-around: příklad zdravotnictví, conformed dimenze pacient



Dimensionální modelování

- BUS architektura
 - Zajišťuje napojení jednotlivých etap (datových tržišť) navzájem a tak zajišťuje vytvoření celistvého DW
 - Datová tržiště by měla obsahovat co nejvíce podrobná atomická data
 - Vytvořena za pomoci Conformed dimensions a standard fact
- Conformed dimensions
 - Dimenze, která znamená to samé ve všech faktových tabulkách na které může být připojena
 - Je stejná ve všech datových tržištích
 - Klasickým příkladem: Zákazník, Produkt, Region, Čas
 - Potřeba identifikovat tyto dimenze, udržovat je, publikovat a dodržovat
 - Data pro conformed dimenzi často z více zdrojů
 - Fyzicky jsou implementovány jednou tabulkou napojitelnou na všechny relevantní faktové tabulky
 - Dovoluje tedy operaci drill across
 - Návrh většinou na co nejdetailnější úrovni
 - Využívat umělé klíče
 - Větší flexibilita
 - Nezávislost na OLTP
 - SCD

Dimensionální modelování

- **Conformed standard fact**
 - Aby zaručilo bezchybné drill across
 - Stejná fakta musí být stejně uložena ve všech datových tržistiích (např. příjem)
 - Stejně měrné jednotky (ne jednou v jednotkách, jednou v krabicích)
 - Lze ztěžít porovnat v reportech
 - Uložit oboje
 - Označuje-li jiné věci, pojmenovat rozdílně
- **Není-li třeba provázanosti, je někdy možné vytvářet nezávislé datové tržiště (např. prodáváme-li rychlé občerstvení a traktory, nemusí mít cenu zákazník sdužovat do jedné dimenze)**

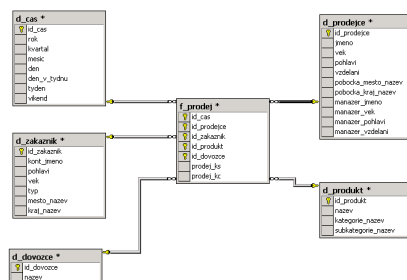
Dimensionální modelování

- **Praktické zkušenosti:**
 - Obvykle 4 – 15 dimenzí na faktovou tabulku
- **Fakt – je něco co není známo většinou předem, co pozorujeme, co vychází z chování trhu, ...**
- **Dimenze**
 - Často záleží na rozhodnutí designera
 - Lze dimenze produkt a obchod
 - Prodávají-li se produkty ve všech obchodech (nezávislé)
 - Nebo jednu dimenzi obchod-produkt
 - Prodávají-li se určité produkty v určitém obchodě (závislé)
 - Taková dimenze je výhodnější – nese informaci o tom kde se co prodává – lze tuto informaci získat přímo prohlížením (browsing) dimenze

Praktický příklad 3

- Vytvořte multidimenzionální model 1. vrstvy datového skladu:
 - Uživatelé chtějí sledovat:
 - Prodej zboží v korunách a kusech
 - Prodeje v jednotlivých dnech
 - Prodeje podle produktů, jejich kategorií a subkategorií
 - Analyzovat využití dopravců – počet přepraveného zboží
 - Analyzovat prodejce
 - Analyzovat prodeje podle jednotlivých zákazníků a skupin zákazníků (dle geografického umístění)

Praktický příklad 3



Dimenzionální modelování

- Modelovací řešení
 - Slowly Changing Dimensions
 - Rychle se měnící dimenze (RCD)
 - Surrogate Keys
 - Pomocné tabulky (vztahy M:N)
 - Složitě hierarchie v dimenzi
 - Degenerovaná dimenze
 - Junk dimension
 - Heterogenní produkty
 - Spojení fakt tabulek
 - Transakční dimenze
 - Audit dimenze
 - Časová dimenze
 - Mnohonásobné měrné jednotky
- Různě měny
 - Intervalový reporting
 - Změny v dimenzi
 - Velké dimenze
 - Causal dimension
 - Role
 - Many Alternate Realities
 - Unity dimension
 - Faktová tabulka bez faktů
 - Vztahy parent-child
 - On-line DW
 - Sledování zákazníka

Slowly Changing Dimensions

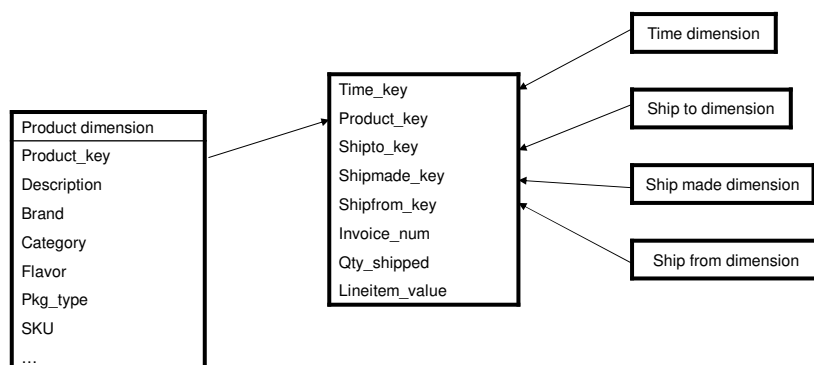
- Problém uchování historie v datech
 - OLTP systémy často neřeší
 - Přepsání hodnot
 - Odmazání historických dat
- Příklad: Změna názvu produktu, ID produktu se nemění
- Tři možnosti řešení
 - Přepsání
 - Vytvoření nového záznamu
 - Vytvoření atributu s aktuální hodnotou

SCD – Typ 1

- Nejednoduší a nejrychlejší
- Neudrží historii
- Stará hodnota pro nás není významná
 - Např. byla špatně zadána

SCD – Typ 2

- Nejpoužívanější technika

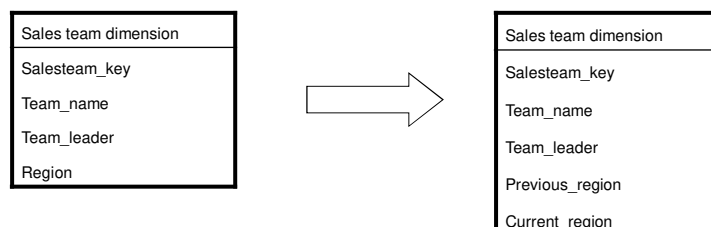


SCD – Typ 2

- Pro každou změnu neklíčového atributu (jehož historii chceme sledovat) se vytvoří nový záznam
- Pro záznamy v dimenzi jsou využity dva klíče
 - IDU – Umělý identifikátor, mění se s změnou atributu, odkazuje do faktové tabulky
 - IDS – konstantní ID
- Je vhodné doplnit do dimenze atributy
 - Platnost od
 - Platnost do
- Výhody/Nevýhody
 - Umožňuje přesně sledovat vývoj v historii
 - Zvětšuje dimenzionální tabulku

SCD – Typ 3

- Vhodné jestliže chceme sledovat vývoj jak podle staré tak nové hodnoty
- Přidají se atributy
 - Aktuální hodnota
 - Předchozí hodnota



Rychle se měnící dimenze (RCD)

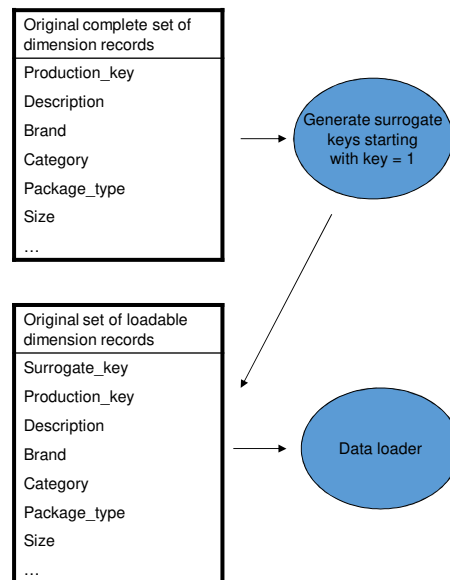
- Velké dimenze, kterých probíhají časté změny
 - Velký nárůst dat
- Např. Dimenze Zákazník a několik atributů scorujících zákazníka (Dobrý, Brzy odejde, ...)
 - Mění se každý měsíc
- Řešení:
 - Jsou-li měnící se atributy textové a používají se pro tvorbu reportů -> Vytvořit pro ně novou dimenzi přímo napojenou na fakt tabulku
 - Jsou-li numerické – převést je na fakta

Surrogate Keys

- Surrogate Keys – umělé klíče, používané v DW místo přirozených primárních klíčů
- Je doporučováno – všechny přirozené klíče nahradit umělými
 - Všechny join do fakt tabulek přes umělé klíče
- Umělé klíče – integer datový typ
 - 4 byte – 2 miliardy hodnot
- Důvody
 - Flexibilní, nezávislé na změnách v OLTP systémech
 - Označení hodnoty „Nevím“ v dimenzi
 - SCD
 - Rychlejší join
 - Většinou menší nároky na místo než u přirozených klíčů
- I časová dimenze by měla mít umělý klíč
 - Nespojovat přes DATE-TIME (datový typ) atribut

Surrogate Keys

- Umělé klíče: přiřazené každé dimenzi
 - Hodnoty 1, 2, 3,
- Přiřazení hodnot umělého klíče:
 - První načtení
 - Následné načtení
 - Využití look-up tabulek



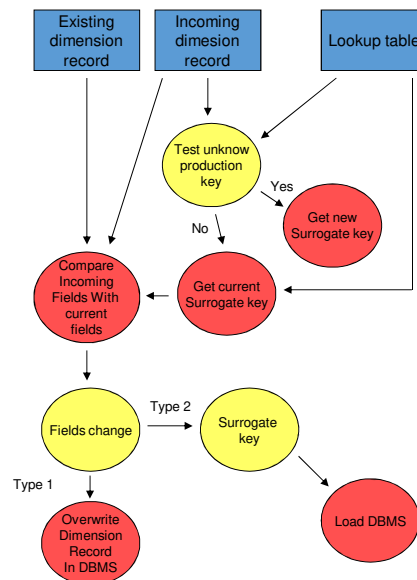
Surrogate Keys

- Look-up tabulka

Production_key	Current_surrogate_key
SKU43WERT567	2345
SKU653TYH7889	4567
SKU34RTB567MM	5436
SKU34ERA23UJ4	2376
...	...

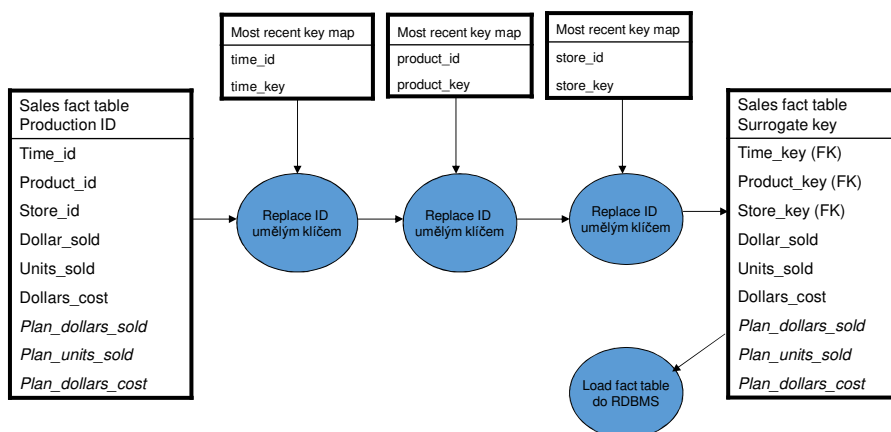
Surrogate Keys

- Nakonec je třeba aktualizovat look-up tabulky
- Jestliže OLTP systémy podporují sledování změn (datum, odpovědnost) nemusí se provádět náročné porovnání položek záznamu.



Surrogate Keys

- Po dimenzích následuje faktová tabulka

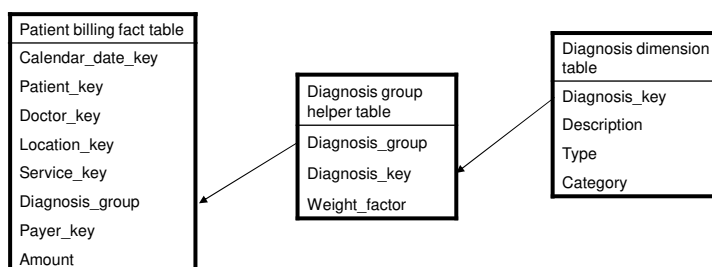


Pomocné tabulky

- Pro řešení vztahu M:N mezi dimenzí a faktovou tabulkou
- Jednomu záznam ve faktové tabulce neodpovídá jeden záznam v dimenzi
- Např.: dimenze diagnóza u sledování pacientů
- Řešení:
 - Vypustit dimenzi
 - Vybrat jednu hlavní hodnotu a ostatní vypustit
 - Vytvořit pevný počet dimenzí – diagnózy (každá dimenze pro jednu diagnózu)
 - Zvětšit granularitu faktové tabulky
 - Využít pomocnou tabulku (Helper table)

Pomocné tabulky

- Pomocná tabulka
- Weight_Faktor zaručuje správné výsledky ve výstupech
 - Fakta jsou násobena Weight_Faktor
 - Součet Weight_Faktor pro jednu skupinu se rovna 1
 - Zde nastavit proporcionálně pro celou skupinu
- Impact report
 - Není využit Weight_Faktor k násobení
 - Nelze potom agregovat

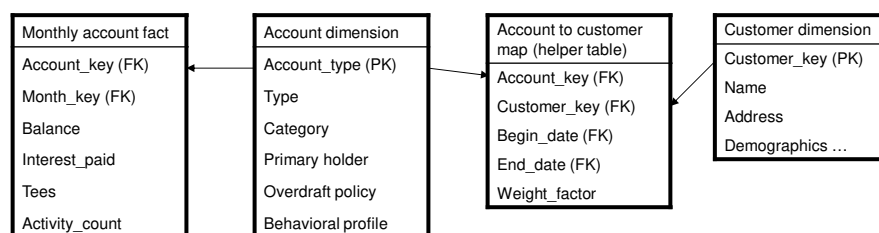


Pomocné tabulky

- Pro uživatele je možné připravit schéma Hvězdy
 - Využit views
 - Spojit pomocnou tabulku na faktovou tabulku
 - Vynásobit fakta Weight_Faktor
 - Výsledek je přímo napojený na dimenzi Diagnóza
- Pomocné tabulky se využijí i v např:
 - Bankovníctví – více vlastníků účtu

Pomocné tabulky

- Příklad:



Pomocné tabulky

- Pomocná tabulka obsahuje umělé klíče do tabulek Klient a Účet
 - Při každé SCD2 změně v některé z těchto tabulek je nutné přidat záznam do pomocné tabulky
 - BeginDate a EndDate v pomocné tabulce ukazují interval v němž byl klient majitelem účtu a obě dimenze se neměnily
- Architektura umožňuje komplexní analýzu
 - S jednoduchými SQL dotazy
- Dopad na složitost ETL: doplnit EndDate do předchozího záznamu

Pomocné tabulky

- Klient účtu ABC123:
 - ```
SELECT customer.name
FROM account, map, customer
WHERE account.accountkey = map.Accountkey
AND customer.customerkey = map.Customerkey
AND account.naturalid = 'ABC123'
AND '7/18/2001' BETWEEN map.begindate AND map.enddate
```

## Pomocné tabulky

- Weight\_Faktor:
  - K alokaci aditivních faktů na jednotlivé položky dimenzionální tabulky
  - Pro daný účet součet roven 1
- Balance na účtu ABC123
  - `SELECT customer.name, fact.balance*weightingfactor`  
`FROM fact, account, map, customer, month`  
`WHERE fact.accountkey = account.Accountkey`  
`AND fact.monthkey = month.Monthkey`  
`AND account.accountkey = map.Accountkey`  
`AND customer.customerkey = map.Customerkey`  
`AND account.naturalid = 'ABC123'`  
`AND month.monthdate = 'July, 2001'`
- Bez faktoru – impact report – pro každého majitele účtu stejná hodnota balance
  - Není možné agregovat na zjištění balance účtu
  - Ukazuje s kolika skutečně penězi může klient disponovat

## Pomocné tabulky

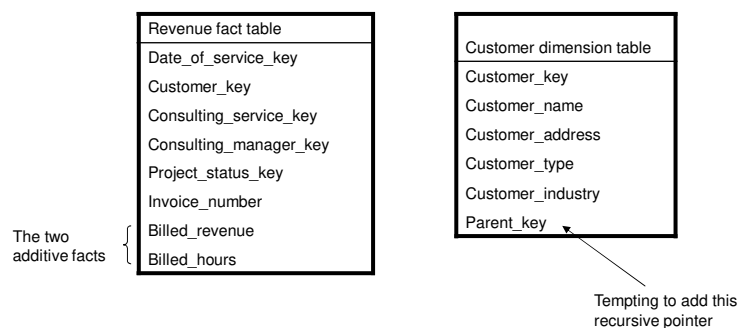
- Pomocná tabulka musí být aktualizovaná:
  - Při změně v dimenzi Účet, Klient
  - Při přidání klienta k účtu, při odebrání klienta z účtu
  - Při úpravě Weight\_Faktor (součet 1)
- Pro zjednodušení je vhodné znovu zapsat všechny klienty pro daný účet do pomocné tabulky (stejný BeginDate)

## Složité hierarchie v dimenzi

- Využití pomocné tabulky pro sledování složitých hierarchií v dimenzi
  - Standardní SQL
- Uvažujme příklad:
  - Konzultantská firma prodává svým zákazníkům
  - Prodeje mohou být provedeny jedné firmě ale na různých úrovních
  - Uživatelé chtějí sledovat jak prodeje celé firmě tak i jednotlivým oddělením

## Složité hierarchie v dimenzi

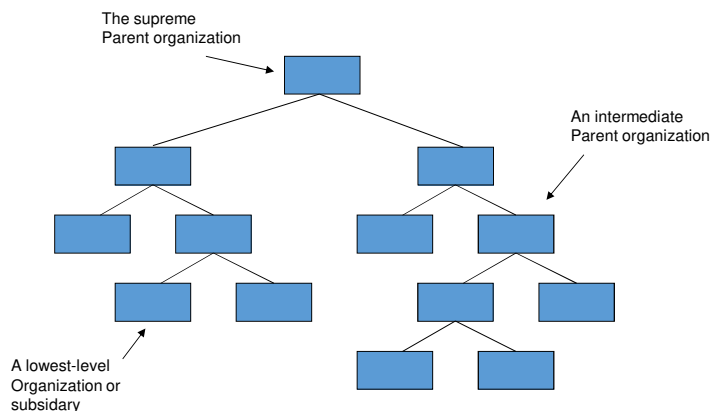
- Klasický rekurzivní atribut (join tabulky zákazníků samu na sebe) neřeší problém:
  - Není možné využít jednoduché SQL pro dotazy





## Složité hierarchie v dimenzi

- Hierarchie

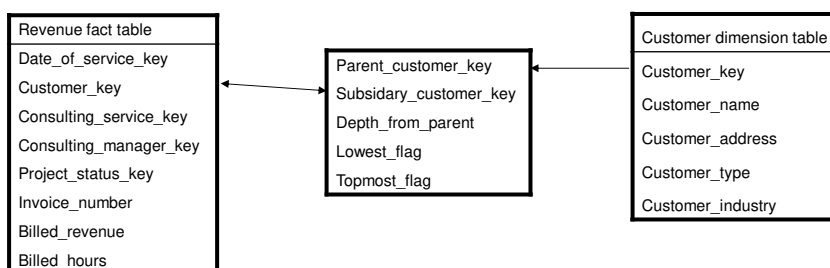


## Složité hierarchie v dimenzi

- Řešení přes pomocnou tabulku
- Nijak neovlivní dosavadní model
- Obsahuje záznam pro každou cestu z uzlu na sebe sama a na uzly pod sebou
- Atributy:
  - Parent Customer Key
  - Subsidiary Customer Key
  - Depth From Parent
  - Lowest Flag
  - Topmost Flag

## Složité hierarchie v dimenzi

- Umožňuje jednoduchým SQL získat pro daný uzel informace o všech potomcích
- Obrácením spojení se můžete dívat na nadřazené, např:
  - Depth = 1 nadřazený
  - Topmost = True - nejvyšší



## Složité hierarchie v dimenzi

- Řešení je možné rozšířit o přidání atributů do pomocné tabulky:
  - Začátek\_Platnosti
  - Konec\_Platnosti
- Umožňuje sledovat změny v hierarchii
- Je-li více rodičů v hierarchii je možné řešit přidáním Weight\_Faktor do pomocné tabulky
  - Součet pro uzel musí být roven 1

## Degenerovaná dimenze

- Degenerovaná dimenze:
  - Většinou se objeví v případě popisu reality objekt – položky objektu (např. objednávka – položky objednávky)
  - Faktová tabulka na úrovni položek objednávky
  - Kam s číslem objednávky?
  - Jedná se o degenerovanou dimenzi – je zapsaná ve faktové tabulce a nemá přímou napojitelnost na žádnou dimenzionální tabulku
  - Vhodná je pro tvorbu reportů, napojení na provozní systémy, ...

## Junk dimension

- Při analýze zdrojových dat zůstanou atributy, které nelze přiřadit k některé dimenzi
  - Např. 10 atributů označujících kód transakce, ....
- Co s nimi?
  - Vytvořit deset dalších dimenzí – může být nepřehledné pro uživatele
  - Spojit všechny atributy do jedné junk dimense
- Junk dimenze – vhodné uskupení náhodných indikátorů a různých atributů
- Pozor aby Junk dimense neobsahovala příliš mnoho záznamů

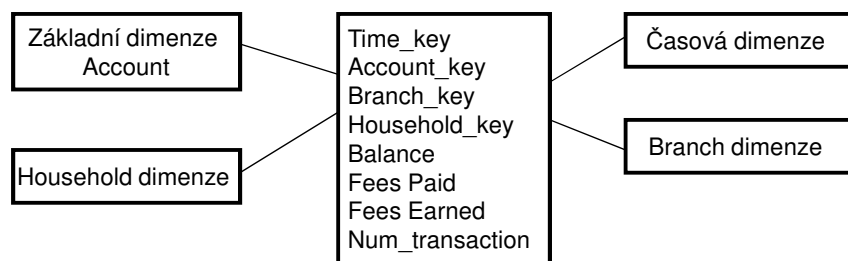
## Junk dimension

- Má-li např. každý atribut 100 různých hodnot a jsou plně nezávislé pak může mít dimenze až  $100^{10}$  záznamů
- Řešení:
  - Rozdělit atributy do skupin korelovaných atributů
  - Např. má-li  $A_1$  100 různých hodnot a  $A_2$  1000 různých hodnot, kolik různých hodnot nabývá kombinace atributů  $A_1$  a  $A_2$ 
    - 1000 –  $A_1$  je rodičem  $A_2$
    - 100 000 –  $A_1$  nezávislé na  $A_2$  – rozdělit do dvou dimenzí
    - 10 000 – lze  $A_1$  do dimenze s  $A_2$
- Nabývají-li např. jen 3 různých hodnot lze vytvořit dimenzi se všemi deseti atributy (maximálně  $3^{10}$  záznamů)

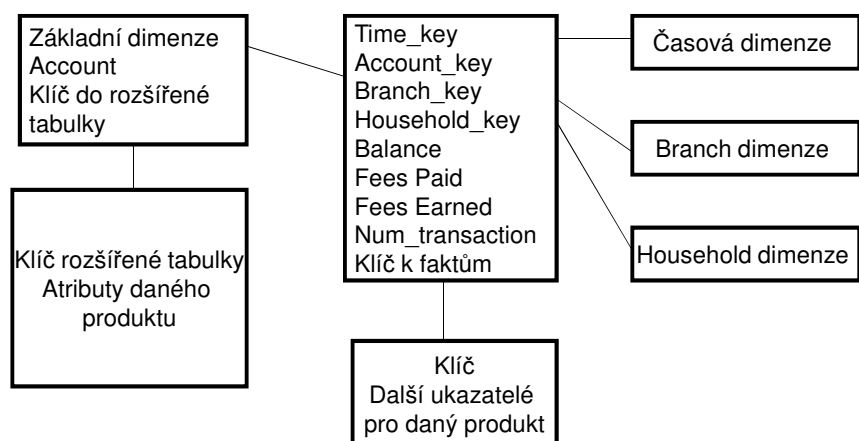
## Heterogenní produkty

- Heterogenní produkty
  - Řešení v businessu kde řada produktů s různými charakteristikami
  - Potřeba sledovat celkový obraz i jednotlivé produkty
- Řešení
  - Základní faktová tabulka – pro všechny produkty
  - Faktové tabulky pro jednotlivé produkty

## Heterogenní produkty



## Heterogenní produkty



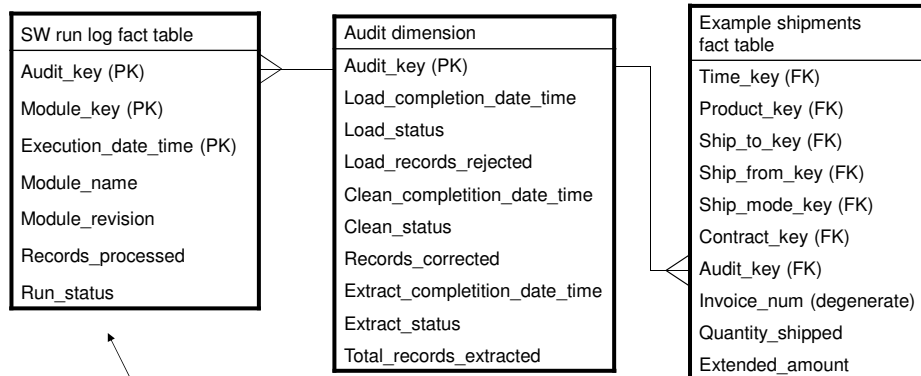
## Spojení fakt tabulek

- Pozor při tvorbě dotazů spojujících jednu dimenzi s více než jednou faktovou tabulkou
  - Není možné použít jednoduchý join

## Audit dimenze

- Pro sledování auditních informací o plnění faktových tabulek
  - Lze rozšířit na vybrané dimenze i celý datový sklad
- V případě že dochází k update faktové tabulky je třeba (vztah M:N mezi faktovou tabulkou a auditní dimenzí) je třeba přidat mezi faktovou tabulku a auditní dimenzi pomocnou tabulku (vazební)

## Audit dimenze



Při potřebě zachytit podrobnější informace

## Časová dimenze

- Vyskytuje se téměř u všech faktových tabulek
- Lze uvažovat dva přístupy
  - Pouze hodnota v faktové tabulce (date-time)
  - Vlastní plnohodnotná dimenze
- Druhá možnost podporuje analýzu
  - SQL funkce nejsou tak komplexní aby zachytily veškerou časovou logiku (např. finanční rok, kalendářní rok, sezónnost, ...)

| Time_key             |
|----------------------|
| Day_of_week          |
| Day_number_in_month  |
| Day_number_overall   |
| Month                |
| Month_number_overall |
| Quarter              |
| Fiscal_period        |
| Season               |
| Holiday_flag         |
| Weekday_flag         |
| Last_day_in_month    |
| ...                  |

## Časová dimenze

- Data jsou plněna „ručně“
  - Např. makro v Excelu
  - Naplnit několik let (10 – 20)
- Většinou granularita denní, v některých případech až na sekundy
  - Vhodné rozdělit, sekundy jako numerická fakta (date-time atribut ve fakt tabulce) + klasická denní časová dimenze

## Časová dimenze

- Základní dimenze v každém datovém skladě
  - Předpokládejme nejprve denní
- Na první pohled jednoduchá, predikovatelná
  - Kalendářní dny, 365 dní za rok
- Vyrůstající otázka okolo časových intervalů
- Příkladem transakce na účtu (otevření účtu, transakce –výběr, uložení, uzavření účtu)
- Otázky bez problémů:
  - Ukaž všechny transakce za daný časový úsek
  - Urči zda daná transakce nastala v daném časovém úseku
  - Umožni pokročilou časovou analýzu při využití informací o svátcích, prázdninách, víkendech, ...
- Předchozí otázky vyžadují klasickou časovou dimenzi
- Vhodné doplnit atributy:
  - Poslední den v časovém intervalu (např. poslední den v čtvrtletí)
  - Hodnoty Ano/Ne



## Čas dne

- Je-li třeba zachytit přesný čas je vhodné ho oddělit od denní časové dimenze
  - Přesný čas je uvažován spíše jako fakt
  - I když může existovat jeho interpretace (čas oběda, čas uzávěrky, ...)
- Potřebujeme-li sledovat více časových zón lze přidat více časových dimenzí

## Čas dne

| Sales fact table     |
|----------------------|
| Date_key (FK)        |
| GMT_data_key (FK)    |
| Product_key (FK)     |
| Customer_key (FK)    |
| Call_center_key (FK) |
| Service_rep_key (FK) |
| Promotion_key (FK)   |
| Time_of_day          |
| GMT_time_of_day      |
| Dollars_sold         |
| Unit_sold            |
| Dollar_cost          |

## Časová dimenze

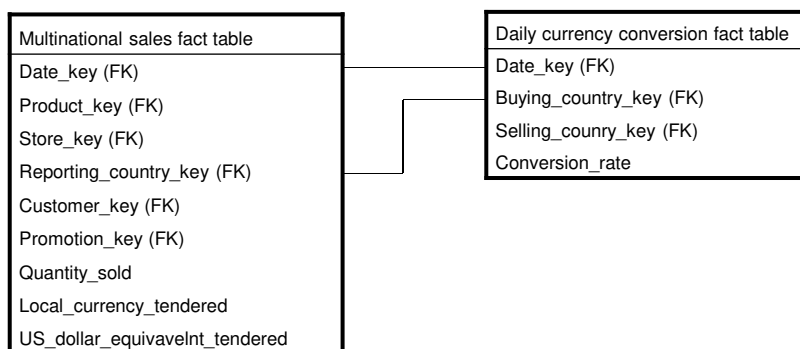
- Složitější situace:
  - Ukaž všechny, kteří jsou zákazníky v daném časovém intervalu.
  - Ukaž poslední transakci zákazníka v daném časovém intervalu
  - Ukaž stav účtu zákazníka v libovolném časovém okamžiku
- K odpovědi vystačíme s jednou časovou dimenzí
  - Dotazy jsou ovšem komplexní a složité
  - Neefektivní dotazy
  - Koncové programy je těžko automaticky definují
- Řešení: Dvě časové dimenze
  - Počátek a konec intervalu
- Odpovědi na otázky:
  - Nalézt všechny účty s cas\_zacatek transakce otevření účtu před koncem intervalu a cas\_konec transakce zavření účtu po začátku daného intervalu
  - Najdi transakci s cas\_zacatek v nebo před koncem intervalu a cas\_konec v nebo po konci intervalu
  - Najdi transakci s cas\_zacatku v nebo před daným okamžikem a cas\_konce v nebo po daném okamžiku
- Vystačíme s BETWEEN klauzulí

## Časová dimenze

- Dopady dvou dimenzí:
  - Musíte navštívit každý záznam v faktové tabulce dvakrát
    - Při prvním insertu (cas\_konce nastaven na neurčito, ne null)
    - Při následujícím insertu na účtu k správnému nastavení předcházejícího cas\_zacatku
- Problém s větší granularitou
  - Lze využít dvě dimenze
  - Nelze využít klasickou předpřipravenou časovou dimenzi – mnoho záznamů (31 milionu sekund za rok)
- Řešení:
  - Čtyři dimenze, dvě s denní granularitou, dvě s sekundovou (pouze date/time ve faktové tabulce)
  - Dynamická časová dimenze (pro daný záznam ve faktové se vygeneruje záznam v časové dimenzi)

## Různě měny

- Pro sledování různých měn
  - Lze uložit fakta v různých měnách
  - Nebo využít speciální faktovou tabulku pro konverzi



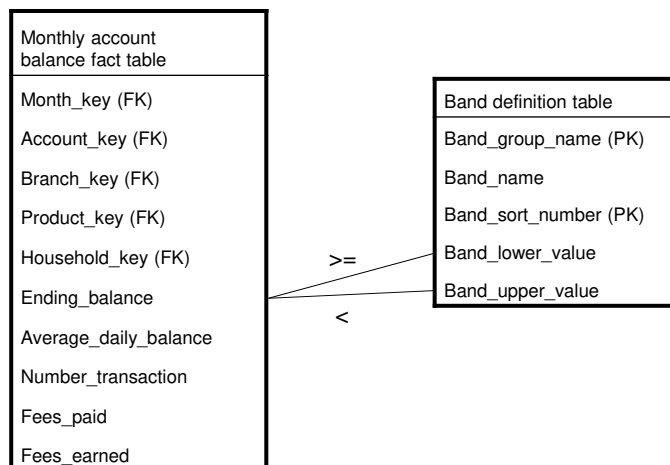
## Intervalový reporting

- Je-li potřeba vytvořit report stylu:

| Balance range | Number of Accounts | Total of Balances |
|---------------|--------------------|-------------------|
| 0 - 1000      | 456                | 123               |
| 1001 - 2000   | 678                | 234               |
| 2001 - 3000   | 345                | 125               |
| 3001 and up   | 123                | 23                |

- Není jednoduché v SQL
  - Možno využít CASE, ...není transparentní
- Lze využít speciální design
  - Join přes spodní a horní meze intervalu
  - Do reportu název intervalu
  - Využívá indexy na faktech

## Intervalový reporting

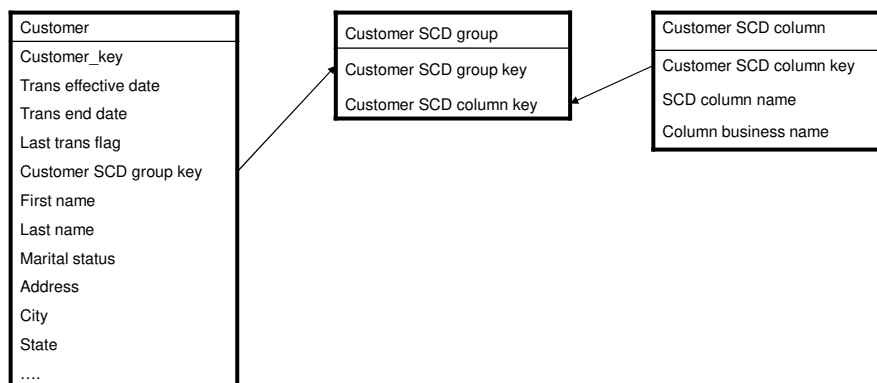


## Změny v dimenzi

- V reálném DW nevystačíme pouze s faktovými a dimenzionálními tabulkami
- Potřeba zachytit změnu v prostředí
- Řešení SCD (typ 2)
- Co když ale chceme prohlížet dimenzi odděleně od faktové tabulky?
  - Kdy se zákazník oženil?
  - Kdy poprvé si potom zakoupil zboží?
  - Které další atributy se u zákazníka změnily?
  - Které atributy se obvykle mění společně?

## Změny v dimenzi

- Řešení:



## Změny v dimenzi

- Customer SCD column mohou zároveň sloužit jako metadata pro vývojáře ETL – co sledovat za změny (typ 2)

| Customer SCD group key | Customer SCD column key |
|------------------------|-------------------------|
| 0                      | 0                       |
| 501                    | 1                       |
| 501                    | 2                       |
| 502                    | 3                       |
| 502                    | 4                       |

| Customer SCD column key | SCD column name | Column business name |
|-------------------------|-----------------|----------------------|
| 0                       | New record      | New record           |
| 1                       | Last_name       | Last name            |
| 2                       | Marital_status  | Marital status       |
| 3                       | City            | City                 |
| 4                       | State           | State                |

| Customer key | Trans effective date | Trans end date | Last trans flag | Customer SCD group key | First name | Last name | Marital status | City     | State |
|--------------|----------------------|----------------|-----------------|------------------------|------------|-----------|----------------|----------|-------|
| 1            | 1.4.2004             | 15.6.2004      | N               | 0                      | Mary       | Smith     | Single         | New York | NY    |
| 2            | 15.6.2004            | 12.7.2004      | N               | 501                    | Mary       | Jones     | Married        | New York | NY    |
| 3            | 12.7.2004            | 31.12.9999     | Y               | 502                    | Mary       | Jones     | Married        | Westport | CT    |

## Změny v dimenzi

- 0 pomáhá identifikovat kdy zákazník byl vložen do systému
- V obrázku je využita technika dvou časových dimenzí `cas_zacatku`, `cas_konce` efektu transakce
- `Last_Trans_Flag` – pro nalezení poslední transakce – aktuálního profilu zákazníka
- Technika je využitelná nejen pro dimenzi zákazník (produkt, dodavatel, zaměstnanec)

## Náročné úlohy

## Velké dimenze

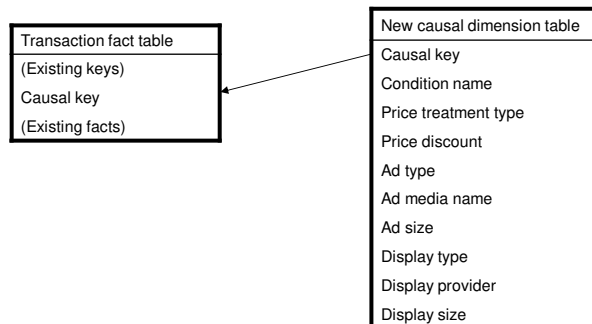
- Některé dimenze mohou být opravdu velké
  - Zákazníci
  - Produkty (500 000)
- Nutné využít bitmapové indexy
- Rozdělit velké dimenze
  - Stabilní část
  - Proměnlivá – SCD (rozdělit na intervaly, naplnit všemi možnostmi)
  - Obě dimenze napojené na fakt tabulku
  - Nebo neimplementovat SCD
- Problémy:
  - Ztráta informace při převodu na intervaly
  - Proměnlivá dimenze nesmí být moc velká – řešení vytvořit jich několik
  - Zpomalení prohlížení dimenze, nutné přes faktovou tabulku
  - Změna pouze při zápisu do faktové tabulky – řešení vytvořit umělý typ transakce změna atributu, kdy se změní atributy v proměnlivé dimenzi

## Causal dimension

- Zachycuje důvody proč došlo k zápisu do faktové tabulky
- Příklad: Prodej v obchodním řetězci
  - Proč zákazník nakoupil
  - Důvody: Platí sleva, promo akce, ochutnávka, ...
- Obdobně telekomunikace: sleva na volání, akce na víkend, nej číslo, ...
- Vhodné nalézt tyto informace a dodat do modelu
  - Podpora marketingových analýz
- Nemá vliv na původní faktovou tabulku
  - Nemění její granualitu

## Causal dimension

- Důležité je nezapomenout na hodnotu: Žádná promoce



## Role

- Výskyt dimenzionální tabulky ve více rolích
- Např.: Akumulovaná fakt tabulka objednávek
  - Dimenze: Datum objednávky, Datum balení, Datum dodávky, Datum platby, ..., Zákazník, Produkt
- Není možné napojit vše na jednu časovou dimenzi
  - Fyzicky je možné udržovat pouze jednu dimenzi
  - Využít view, kopie, materializované view – každá dimenze své jméno (i atributy)
- Podobně je možné řešit např. výskyt Regionu v různých dimenzích (Zákazník, Dodavatel, ...)
  - Pouze jedna dimenze Region
  - Napojení přes Sněhovou vločku na dimenze ostatní