IIT School of Applied Technology
ILLINOIS INSTITUTE OF TECHNOLOGY
information technology & management

526 Data Warehousing

April 2, 2019
Week 11 Presentation

Week 11 Topic: Dimensional Modeling: More Dimension Patterns and Considerations

- We will cover
  - Dealing with NULL
  - Junk Dimension
  - Outriggers
  - Monster Dimension
  - Dimension Value Band
  - Bridge Tables

Dealing with Nulls

- NULL dimension attributes
  - Strongly discouraged to avoid unexpected query results (e.g. invalidating index strategy)
  - Use default values instead – N/A, Unknown, Invalid, To be determined,…
- NULL facts
  - Use ONLY IF it truly means N/A, Unknown, and Invalid, not zero

Dealing with Nulls: Operations with Nulls

- Operations with Null can be tricky sometimes

```
SELECT 1 + null col
  FROM dual; -- null

SELECT CASE WHEN null = null THEN 1 ELSE 2 END AS col
  FROM dual; -- 2

SELECT CASE WHEN null IS null THEN 1 ELSE 2 END AS col
  FROM dual; -- 1

SELECT CASE WHEN 1 IS null THEN 1 ELSE 2 END AS col
  FROM dual; -- 2

SELECT DECODE (null, null, 1, 2, 3) as col
  FROM dual; -- 1

SELECT DECODE (1, null, 1, 2, 3) as col
  FROM dual; -- null
```

Dealing with Nulls: Operations with Nulls (Cont'd)

- Operations with Null can result in missing information

```
SELECT deptid, SUM(annual_salary) AS annual_salary_by_dept
  FROM (
        SELECT deptid, empid, month_pay*12+bonus AS annual_salary
          FROM (
                SELECT 'd01' AS deptid, 'e0001' AS empid,
                       10000 AS month_pay, 1000 AS bonus
                  FROM dual UNION ALL
                SELECT 'd01' AS deptid, 'e0002' AS empid,
                       100000 AS month_pay, null AS bonus
                  FROM dual
               )
       )
 GROUP BY deptid
;

DEPTID ANNUAL_SALARY_BY_DEPT
------ ----------------------
d01                   121000 -- $1,200,000 missing!
```


Dealing with Nulls: Operations with Nulls (Cont'd)

- Aggregate functions handle Null gracefully

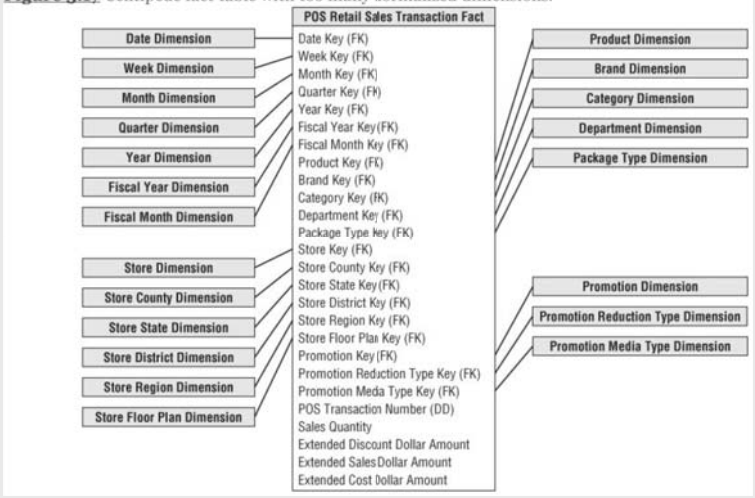
```
SELECT room_no, AVG(math) AS avg_math_by_room, AVG(writing) AS
avg_writing_by_dept
  FROM (
        SELECT room_no, student_id, math, writing
          FROM (
                SELECT 'rm2001' AS room_no, 's0001' AS student_id,
                        80 AS math,   90 writing
                  FROM dual UNION ALL
                SELECT 'rm2001' AS room_no, 's0002' AS student_id,
                       100 AS math, null writing
                  FROM dual
               )
       )
 GROUP BY room_no;

ROOM_NO AVG_MATH_BY_ROOM AVG_WRITING_BY_DEPT
------- --------------- -------------------
rm2001               90                  90
```


Dealing with Nulls: Rule of Thumb in Data Warehousing

- NULL fact table foreign keys
  - No NULL is allowed as it breaks referential integrity
  - Substitute key to special dimension row (a.k.a. dummy dimension row)
- NULL dimension attributes
  - Strongly discouraged to avoid unexpected query results (e.g. invalidating index strategy)
  - Use default values instead – N/A, Unknown, Invalid, To be determined,…
- NULL facts
  - Use ONLY IF it truly means N/A, Unknown, and Invalid, not zero

Avoid Too Many Dimension "Centipede" Fact Tables

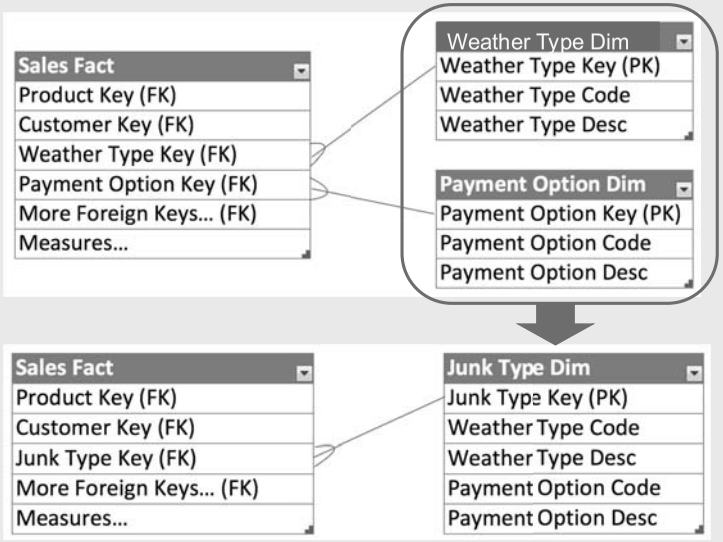
Figure 3.17 Centipede fact table with too many normalized dimensions.



Junk Dimensions

- Combine miscellaneous transaction flags/indicators into junk dimension
  - SCD Type 2 is not necessary
- Potentially less desirable alternative:
  - Multiple fact table keys to low-cardinality dimensions
- Undesirable alternatives:
  - Place flags/indicators directly in fact table as text facts or DDs

Junk Dimensions (cont'd)

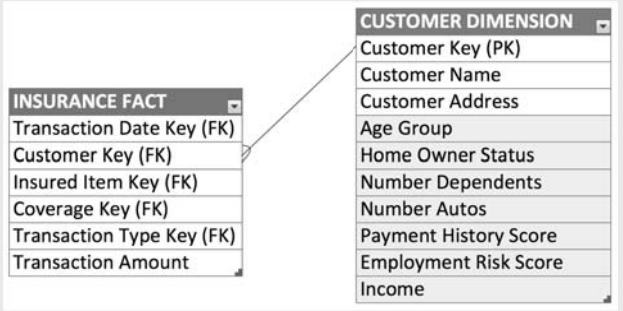


Outriggers

- Dimension tables joined to other dimension tables
- In this case, a date dimension serves as an outrigger to the employee dimension via role-playing
- Outriggers are acceptable in moderation but should be viewed as the exception rather than the rule

Dealing with Rapidly Changing Monster Dimensions: Monster Dimensions

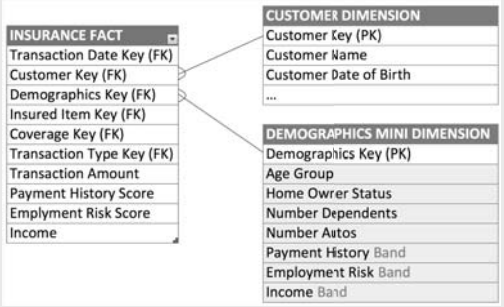
- Imagine an insurance company with a big customer dimension (e.g. 30 million) with rapidly changing demographics (in green)



- The dimension table size can be easily doubled within a short period making this a rapidly changing monster dimension

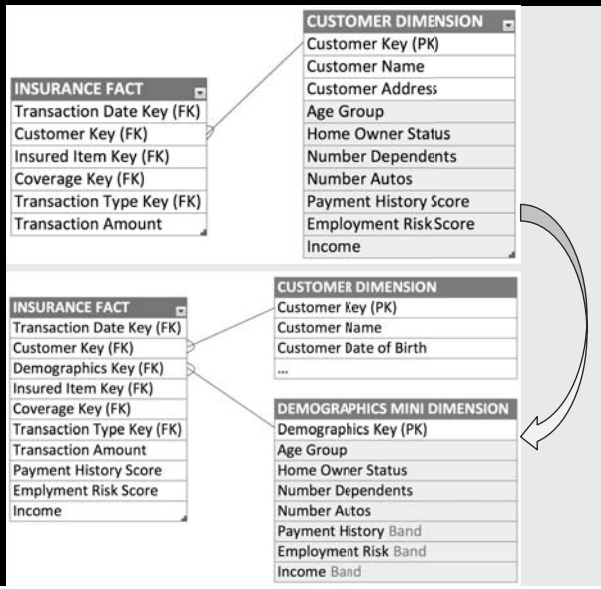
Dealing with Rapidly Changing Monster Dimensions: Monster Dimensions (cont'd)

- The solution is to break off the hot attributes into their own separate mini dimension



- The mini dimension contains one row for each possible combination of the attributes
- Value bands are used in the mini-dimension to reduce the number of rows overall

Dealing with Monster Dimensions: Mini-Dimension to the Rescue



Break off the hot attributes into their own separate mini dimension

It has one row for each possible combination of the attributes

Value bands are used to reduce the number of rows overall

Dealing with Rapidly Changing Monster Dimensions: Monster Dimensions (cont'd)

Customer dimension sample row:

Customer Key	Customer Name	Date of Birth
123456	John Smith	1984-02-10

Demographics mini-dimension sample row:

Demographics Key	Age Group	Income Band
1	25-29	$50,000 - $59,999
2	30-34	$50,000 - $59,999
3	30-34	$60,000 - $69,999

Fact table sample row:

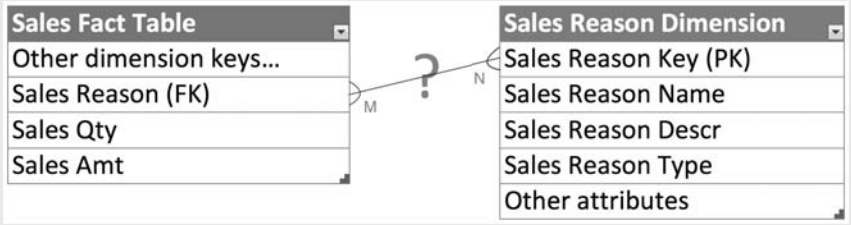
Transaction Date Key	Customer Key	Demographics Key
20140131	123456	1
20140228	123456	2
20140331	123456	2
20140430	123456	3

Dimension Value Band Example

Sugar Level	Daily Calories Burned	Age Group
50-65	0-500	0 to 10
66-80	501-1000	11 to 20
81-95	1001-1500	21 to 30
96-100	1501-2000	31 to 40
101-115	2001-2500	41 to 50
116-130	2501-3000	51 to 60
131-145	3001-3500	61 to 70
146-160	3501-4000	71 to 80

Resolving Multivalued Relationships using Bridge Tables

- In a classic dimensional schema, each dimension attached to a fact has a single value consistent with the fact table's grain
- But there are a number of situation in which a dimension is legitimately multivalued



Resolving Multivalued Relationships using Bridge Tables: Examples

- Many sales reasons on a single transaction
- Many customers in a bank account
- Many diagnosis at a time of treatment
- Many witnesses to an accident
- Many options on a car

Resolving Multivalued Relationships using Bridge Tables
Multivalued Sales Reasons Bridge

Sample rows from Sales Reason Group

Sales Reason Group Key	Sales Reason Group Name
101	Product Quality
102	Product Quality; Promotion
103	Product Quality; Convenient Location

Sample rows from Sales Reason Bridge

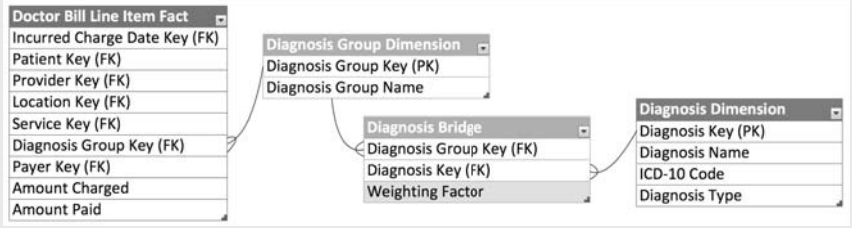
Sales Reason Group Key	Sales Reason Key	Weighting Factor
101	1	1
102	1	0.5
102	2	0.5
103	1	0.5
103	3	0.5

Sample rows from Sales Reason Dimension

Sales Reason Key	Sales Reason Name	Other attributes
1	Product Quality	...
2	Promotion	...
3	Convenient Location	...

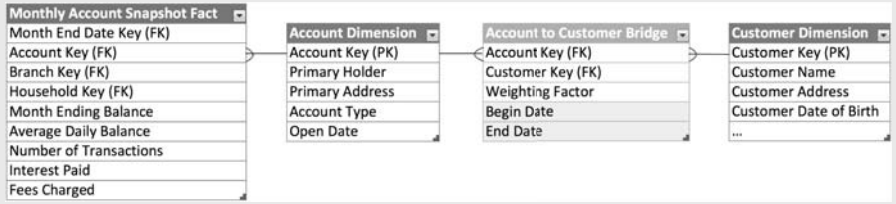
The Sales Reason Group table may be required by your modeling tool to resolve FK/PK relationships. It provides no useful information at query time and is often omitted.

Resolving Multivalued Relationships using Bridge Tables
Multivalued Diagnosis Bridge



- The weighing factor is an explicit allocation
- Records in the Diagnosis Group Dimension can be made for each patient, but in this case it seems reasonable to re-use diagnosis groups, especially for out patient treatments where many groups would be repeated

Resolving Multivalued Relationships using Bridge Tables
Multivalued Bank Account Customer Bridge



- Associate customers to accounts where these have a many-to-many relationship
- Query account balances by individual customer or groups of customers
- Show account balances correctly weighted (prorated) by individual customers to avoid double counting

Week 11 Topic: Dimensional Modeling: More Dimension Patterns and Considerations

Questions?