



您的信息 您的智慧

2011 IBM 信息管理与业务分析论坛





您的信息 您的智慧

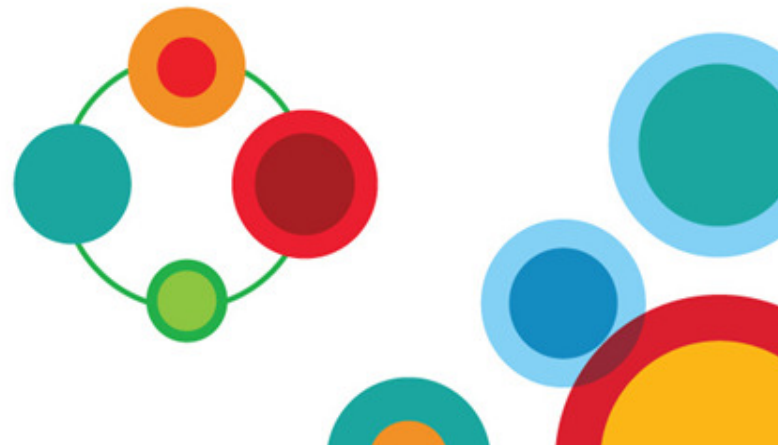
2011 IBM 信息管理与业务分析论坛

DB2 前沿技术主题

Gene Fu, Ph.D.

IBM 杰出工程师

IBM 中国开发中心信息管理首席技术官



免责声明

© 版权所有 IBM Corporation [当年]。保留所有权利。

美国政府用户受限的权利— 使用、复制或泄密受 GSA ADP 与 IBM 公司签署的计划合同的限制。

本演示文稿中所包含的信息只用于提供信息的目的。虽然在检查本演示文稿信息时尽量保证其完整性和准确性，但它只根据“现状”提供，没有任何隐含或者明确的担保。此外，本文包含的信息根据 IBM 当前产品计划和策略提供，如有变更，恕不通知。IBM 不承担因为使用本演示文稿内容和相关内容而造成损害的责任。本演示文稿中包含的内容不打算、也不应该作为 IBM 或其供应商或其许可证销售商的担保或表示，或者修改适用于 IBM 软件的许可证协议的条款和条件。

IBM、IBM 徽标、ibm.com、DB2 和 z/OS 是国际商业机器公司在美国和/或其他国家（地区）的商标或注册商标。如果这些和其他 IBM 商标在本文中第一次出现时标记了商标符号（® 或 ™），均代表在本文出版之际，它们是 IBM 在美国或其他国家注册的商标或普通法规定的商标。此类商标在其他国家（地区）也可能是注册商标或普通法规定的商标。可在网络上获取 IBM 商标的最新列表，请查看 ibm.com/legal/copytrade.shtml 的“Copyright and trademark information”部分。

议程

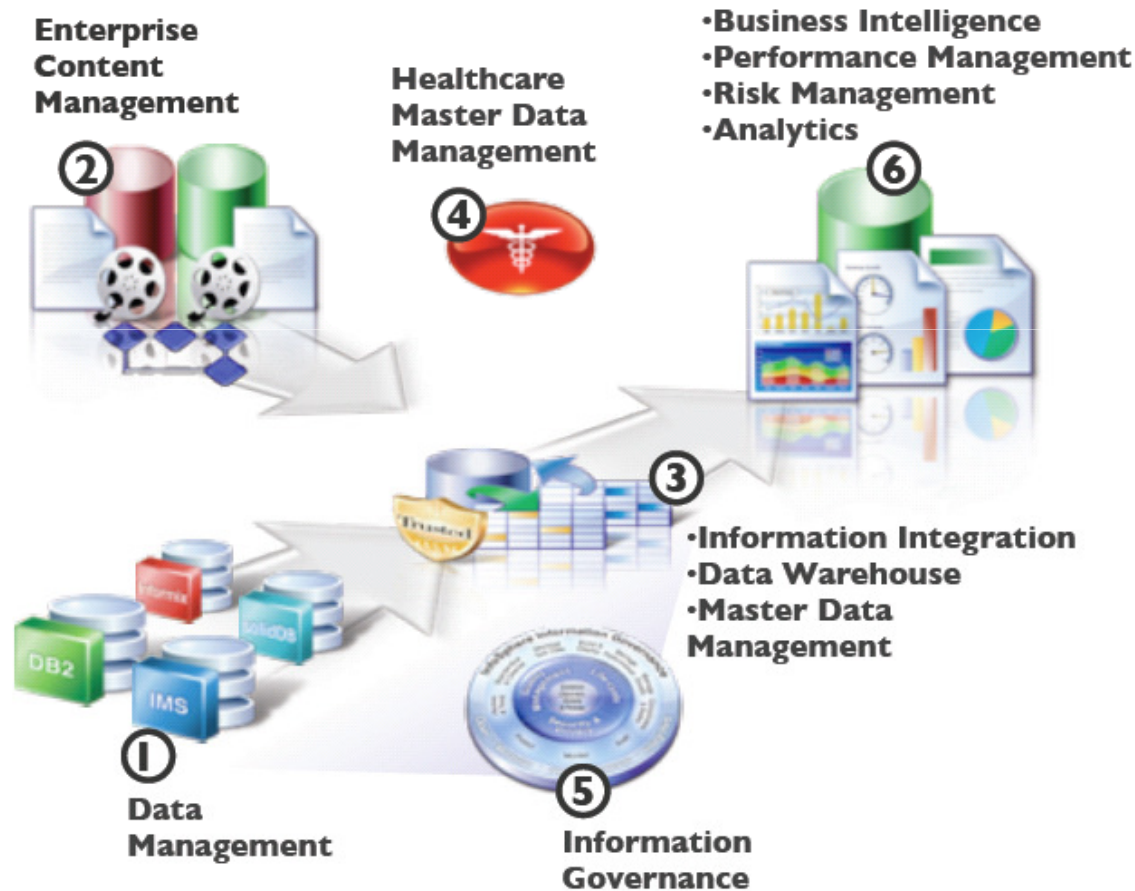
- IBM Information Management
- 双时态数据库（BiTemporal Database）
 - 用例和业务场景
 - SQL 扩展
 - 示例
- 细粒度访问控制
 - 用例和业务场景
 - SQL 扩展
 - 示例

议程

- IBM Information Management
- 双时态数据库（BiTemporal Database）
 - 用例和业务场景
 - SQL 扩展
 - 示例
- 细粒度访问控制
 - 用例和业务场景
 - SQL 扩展
 - 示例

IBM Information Management

IBM Information Management *Selected Key Investments*



Investments

①	IMS 1966	DB2 1983	
	Informix Relational DB 2001	Solid In memory DB 2008	
②	FileNet Content & business process management 2006	Datacap Data & document capture 2010	
	Trigo Product info management 2004	DWL Customer data integration 2005	Ascential Enterprise data integration 2005
③	SRD Identity resolution 2005	Unicorn Metadata management 2006	DataMirror Real time data integration 2007
	Netezza Data warehousing, 2010		
	Initiate Healthcare master data management 2010		
④	Princeton Softech Archiving & test data management 2007	Exeros Data discovery 2009	Guardium Database security 2009
	Cognos BI & PM 2008	SPSS Predictive analytics 2009	OpenPages Risk management 2010

Sample Technology Focus in Data Servers

- pureXML
- Spatial database
- Autonomic computing
- SSD's and Storage Optimizations
- DB2 pureScale
- Big Insights
- DBaaS and Multi-Tenant Database
- BiTemporal Database
- Fine Grain Access Control
- And many more ...



议程

- IBM Information Management
- 双时态数据库（BiTemporal Database）
 - 用例和业务场景
 - SQL 扩展
 - 示例
- 细粒度访问控制
 - 用例和业务场景
 - SQL 扩展
 - 示例

客户提出的双时态数据库支持要求

- 货币中心银行的 CTO：我有 40% 的时间用来处理法规遵从性
 - 可转换为系统时间支持和系统时间的版本控制
- 保险公司 CIO：我们将支持遵从性所需的应用程序打包在一起——希望能够记录所有数据库更改，而不必更改打包的应用程序。
 - 可转换为通过系统时间的双表方法保留应用程序
- 加拿大银行的首席技术人员：“我们在 20 年前就来到你们的实验室，提出了这样的要求，这些年你们去了哪里？”
 - 可转换为
 - 允许将现有表更改为时态表，将包含历史数据的现有表更改为包含历史数据的 DB2 表
 - 允许普通 SQL 处理时态表和新 SQL
- 华尔街领先投资银行的副总裁——从历史记录中清除数据并非基于严格时间的流程，但错综复杂
 - 可转换为基于 SQL 的历史数据删除（经由特殊授权）

提案摘要

- 业务时间（生效日期、有效时间、起始/终止日期）
 - 每一行都有一对时间戳（日期），由应用程序设置
 - 开始时间：业务认为该行有效的时间
 - 结束时间：业务认为该行有效性已终止的时间
 - 在业务时间的当前、任意此前或未来的时间点/时期进行查询
- 系统时间（断言日期、认知日期、事务处理时间、审计时间、传入/发出日期）
 - 每一行都有另一对由 **DBMS** 设置的时间戳
 - 开始时间：该行插入 **DBMS** 的时间
 - 结束时间：该行被修改/删除的时间
 - 修改后的行的开始时间是修改时间
 - 在系统时间的当前或任意此前时间点/时期进行查询
- 双时态（二维历史，二维里程碑）
 - 在行中同时包含系统时间和业务时间

系统时间用例

产品表

步骤	日期	活动
1	6/15/2007	新产品引入
2	6/15/2008	产品降价
3	9/15/2008	产品停产

DML CREATE TABLE PRODUCTS
(Prodname VARCHAR (40),
Price INTEGER
);

Prodname	Price
----------	-------

第 1 步 INSERT INTO PRODUCTS
VALUES ('IMagic', '750')

Prodname	Price
IMagic	750

第 2 步 UPDATE PRODUCTS
SET Price=Price - 300
WHERE Prodname='IMagic'

Prodname	Price
IMagic	450

第 3 步 DELETE FROM PRODUCTS
WHERE Prodname = 'IMagic'

Prodname	Price
----------	-------

可通过保留更新/删除的行来满足大部分合规要求。

- | 步骤 | 日期 | 活动 |
|----|-----------|-------|
| 1 | 6/15/2007 | 新产品引入 |
| 2 | 6/15/2008 | 产品降价 |
| 3 | 9/15/2008 | 产品停产 |

CREATE TABLE PRODUCTS

(Prodname VARCHAR (40),
 Price INTEGER,
 SysTmSta **TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN**,
 SysTmEnd **TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END**, ...
PERIOD SYSTEM_TIME (SysTmSta, SysTmEnd));

Prodname	Price	SysTmSta	SysTmEnd
----------	-------	----------	----------

INSERT INTO PRODUCTS **FOR PORTION OF SYSTEM_TIME**
FROM TRANSACTION TIME TO 12/30/9999

6/15/2007

VALUES ('IMagic', '750',default,default)



Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	12/30/9999

UPDATE PRODUCTS **FOR PORTION OF SYSTEM_TIME**
FROM TRANSACTION TIME TO 12/30/9999

SET Price=Price-300 WHERE Prodname='IMagic'

6/15/2007

6/15/2008



Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	6/15/2008
IMagic	450	6/15/2008	12/30/9999

DELETE FROM PRODUCTS **FOR PORTION OF SYSTEM_TIME**
FROM TRANSACTION TIME TO 12/30/9999

WHERE Prodname = 'IMagic'

6/15/2007

6/15/2008

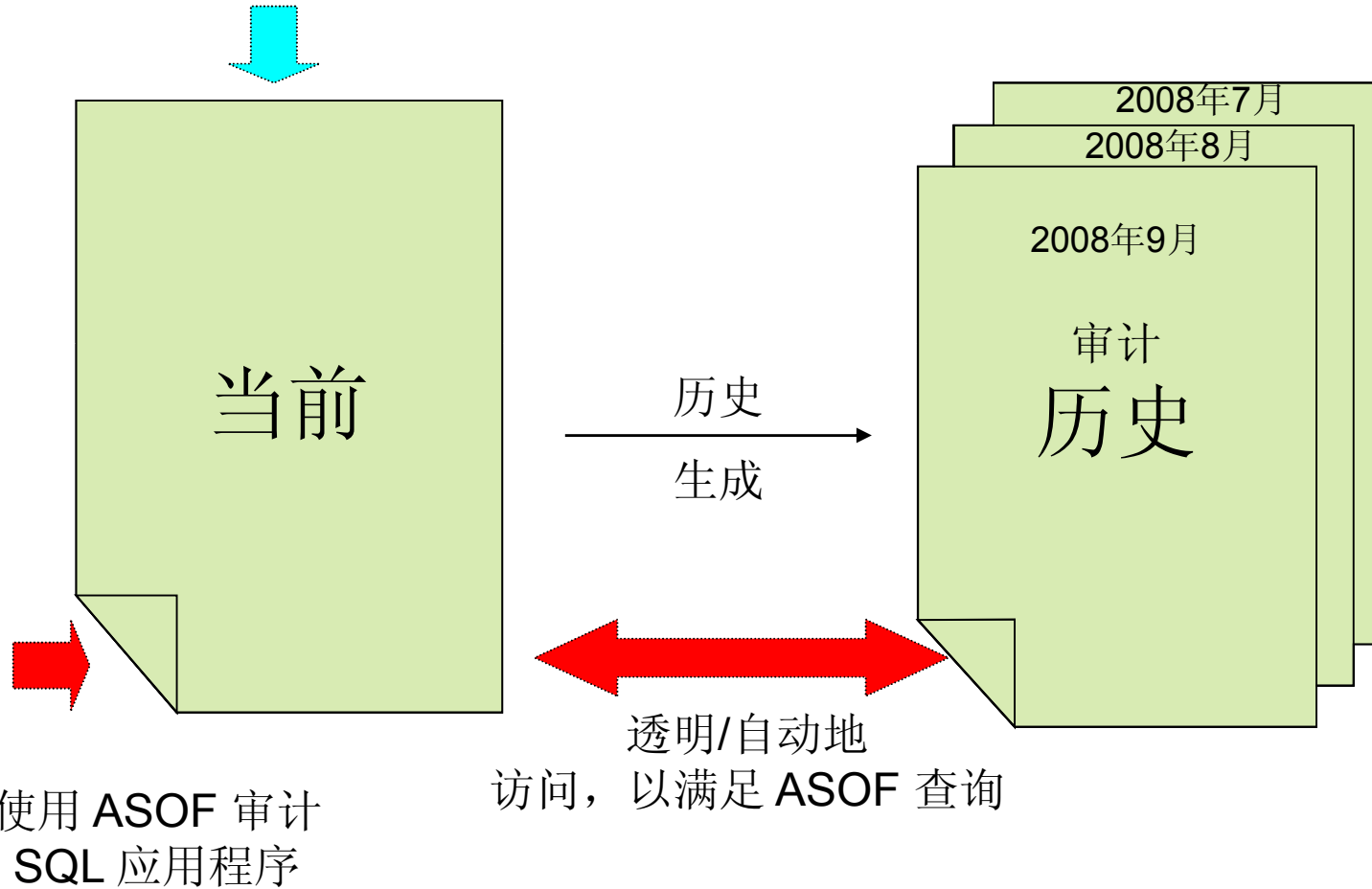
9/15/2008



Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	6/15/2008
IMagic	450	6/15/2008	9/15/2008

当前和历史

当前 SQL 应用程序



```
CREATE TABLE PRODUCTS
(Prodname VARCHAR (40),
Price INTEGER,
SysTmSta TS(12) NN GENERATED ALWAYS AS ROW BEGIN,
SysTmEnd TS(12) NN GENERATED ALWAYS AS ROW END, ..
PERIOD SYSTEM_TIME (SysTmSta, SysTmEnd));
```

```
CREATE TABLE PRODHIST
(Prodname VARCHAR (40),
Price INTEGER,
SysTmSta TS(12)
SysTmEnd TS(12) .. )
;
```

步骤	日期	活动
1	6/15/2007	新产品引入
2	6/15/2008	产品降价
3	9/15/2008	产品停产

```
ALTER TABLE PRODUCTS ADD VERSIONING USE HISTORY TABLE PRODHIST
```

Prodname	Price	SysTmSta	SysTmEnd
----------	-------	----------	----------

```
INSERT INTO PRODUCTS FOR PORTION OF SYSTEM_TIME
FROM DATE TRANSACTION TIME TO 12/30/999
VALUES ('IMagic', '750')
```

Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	12/30/9999

当前行

```
UPDATE PRODUCTS FOR PORTION OF SYSTEM_TIME
FROM DATE TRANSACTION TIME TO 12/30/999
SET Price=Price - 300
WHERE Prodname='IMagic'
```

Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	6/15/2008
IMagic	450	6/15/2008	12/30/9999

历史行

当前行

```
DELETE FROM PRODUCTS FOR PORTION OF SYSTEM_TIME
FROM DATE TRANSACTION TIME TO 12/30/9999
WHERE Prodname = 'IMagic'
```

Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	6/15/2008
IMagic	450	6/15/2008	9/15/2008

历史行

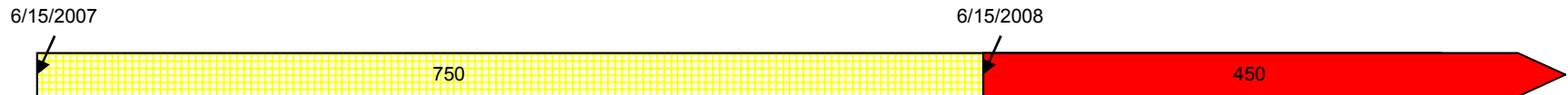
历史行

- | 步骤 | 日期 | 活动 |
|----|-----------|-------|
| 1 | 6/15/2007 | 新产品引入 |
| 2 | 6/15/2008 | 产品降价 |

Prodname	Price	SysTmSta	SysTmEnd
IMagic	750	6/15/2007	12/30/9999

UPDATE PRODUCTS FOR PORTION SYSTEM_TIME
 FROM DATE TRANSACTION TIME TO 12/30/9999
 SET Price=Price-300
 WHERE Prodname='IMagic'

历史行	Prodname	Price	SysTmSta	SysTmEnd
	IMagic	750	6/15/2007	6/15/2008
当前行	IMagic	450	6/15/2008	12/30/9999



当前时间, 6/30/2008
 SELECT Price from PRODUCTS
 FOR SYSTEM_TIME AS OF CURRENT TIME
 WHERE Prodname='IMagic'

Price
450

之前的时间点
 SELECT Price from PRODUCTS
 FOR SYSTEM_TIME AS OF '1/15/2008'
 WHERE Prodname='IMagic'

Price
750

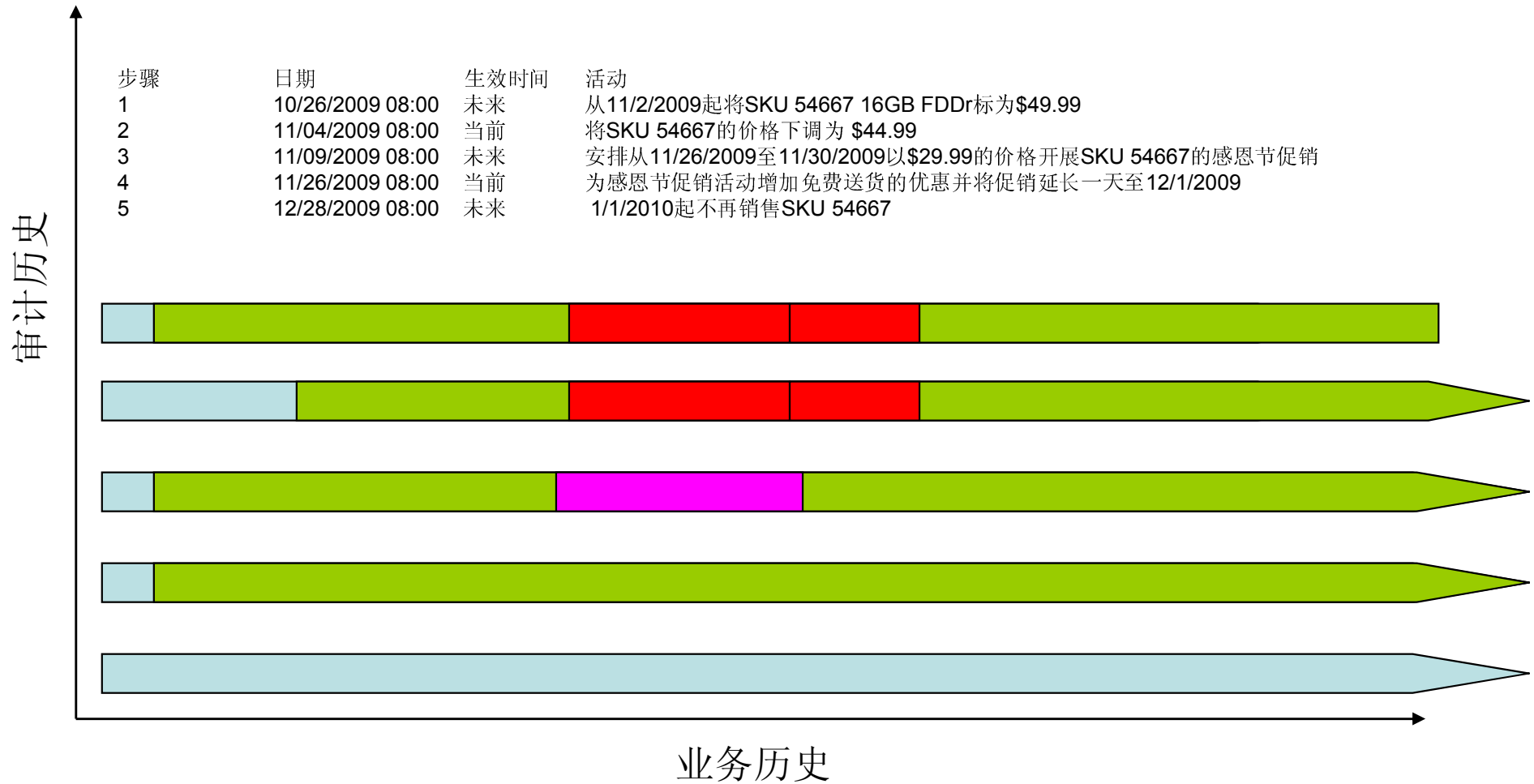
简要重述：DB2 系统时间

- 当前日期的应用得到保留
 - 插入、更新、删除、选择不会更改
 - 触发器不会更改
 - 约束不会更改
- DBMS 通过将历史行移动到独立的表中保留当前应用
- 对当前表执行不包含 AS OF 的 SELECTS
- 对于 SELECTS FST AS OF, DB2 添加必要的日期预测
 - 也支持 SELECT FST FROM/TO、BETWEEN
- DBMS 将 AS OF、FROM/TO、BETWEEN 时间转为系统时间列对的恰当预测
- INSERT/UPDATE/DELETE 的行为方式就像 **FOR PORTION OF SYSTEM_TIME FROM TRANSACTION TIME TO '12/31/9999'** 透明地添加了自动历史创建功能一样
- 常规和此前时间点的查询将使用当前模式名称
- DB2 优化器会在必要时访问当前和历史表，并将结果显示为一个整合的应答集

客户的声音及其如何转换为双时态特性

- 华尔街领先投资银行的副总裁：“几乎所有交易都得到了正确的记录，但在某些时候，我们会发现有些交易者不正确地进入了交易。我们需要能够审计变更，同时对客户的职位进行有追溯力的更新。我们需要实现数据的二维墓碑（tombstoning）：业务历史和审计历史”
 - 可转化为双时态支持
- 另一家华尔街领先投资银行的总经理：“我们需要实现乐观并发控制、可重复的清洁读取、处理未来定位和回溯性的纠正”
 - 阐明双时态支持的需要
- 大型保险公司的首席应用程序开发人员：“我们要求审计历史能够捕获和纠正错误，我们的遵从性规定要求我们保留所发出的所有保单的进展历史。”
 - 强调双时态支持的需要

双时态用例 零售



DDL: 双时态/零售

```
CREATE TABLE Lstng
(SKU VARCHAR(10) NOT NULL,
Desc VARCHAR(20),
Price VARCHAR(10),
Ship VARCHAR(5),
From_D DATE NOT NULL,
To_D DATE NOT NULL,
Asr_Beg TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW BEGIN,
Asr_End TIMESTAMP(12) NOT NULL GENERATED ALWAYS AS ROW END, ..
PERIOD BUSINESS_TIME (From_D, To_D),
PERIOD SYSTEM_TIME (Asr_Beg, Asr_End),
PRIMARY KEY (SKU, BUSINESS_TIME WITHOUT OVERLAPS)
);
```

```
CREATE TABLE LstngHistory LIKE Lstng;
```

```
ALTER TABLE Lstng
ADD VERSIONING
USE HISTORY TABLE LstngHistory;
```

SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
-----	------	-------	------	--------	------	---------	---------

双时态用例： 零售 - DML

步骤	日期	生效日期	活动
1	10/26/2009 08:00 (未来)		从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99

SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
-----	------	-------	------	--------	------	---------	---------

```
INSERT INTO Lstng(SKU, Desc, Price, Ship, From_D, To_D)
VALUES ('54667','16GB FDDr','49.99', 'UPS','11/2/2009','12/31/9999')
```

11/2/2009



SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/09 8:00	12/30/9999 23:59

双时态用例： 零售 - DML

步骤	日期	生效日期	活动
1	10/26/2009 08:00	(未来)	从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99
2	11/4/2009 08:00	(当前)	将SKU 54667的价格下调为 \$44.99

SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	12/30/9999 23:59



```
UPDATE Lstng FOR PORTION OF BUSINESS_TIME
FROM CURRENT DATE TO '12/31/9999'
SET Price='44.99'
WHERE SKU='54667';
```



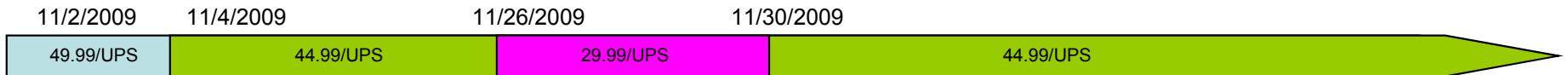
SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	12/30/9999 23:59

步骤	日期	生效日期	活动
1	10/26/2009 08:00	(未来)	从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99
2	11/4/2009 08:00	(当前)	将SKU 54667的价格下调为 \$44.99
3	11/9/2009 08:00	(未来)	从11/26/2009至11/30/2009将SKU 54667的价格调整为\$29.99

SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	12/30/9999 23:59



UPDATE Lstng FOR PORTION OF BUSINESS_TIME
 FROM DATE '11/26/2007' TO '11/30/2007'
 SET Price='29.99' WHERE SKU='54667' ;



SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/31/9999 23:59
54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	11/9/2009 8:00
54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009	11/9/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	29.99	UPS	11/26/2009	11/30/2009	11/9/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	44.99	UPS	11/30/2009	12/31/9999	11/9/2009 8:00	12/30/9999 23:59

- 1 10/26/2009 08:00 (未来) 从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99
- 2 11/4/2009 08:00 (当前) 将SKU 54667的价格下调为 \$44.99
- 3 11/9/2009 08:00 (未来) 安排从11/26/2009至11/30/2009以\$29.99的价格开展SKU 54667的感恩节促销

4 11/26/2009 08:00 (当前) 为感恩节促销活动增加免费送货的优惠并将促销延长一天至12/1/2009

	SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
	54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
	54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/31/9999 23:59
	54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	11/9/2009 8:00
	54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009	11/9/2009 8:00	12/30/9999 23:59
	54667	16GB FDDr	29.99	UPS	11/26/2009	11/30/2009	11/9/2009 8:00	12/30/9999 23:59
	54667	16GB FDDr	44.99	UPS	11/30/2009	12/31/9999	11/9/2009 8:00	12/30/9999 23:59

11/2/2009 11/4/2009 11/26/2009 11/30/2009



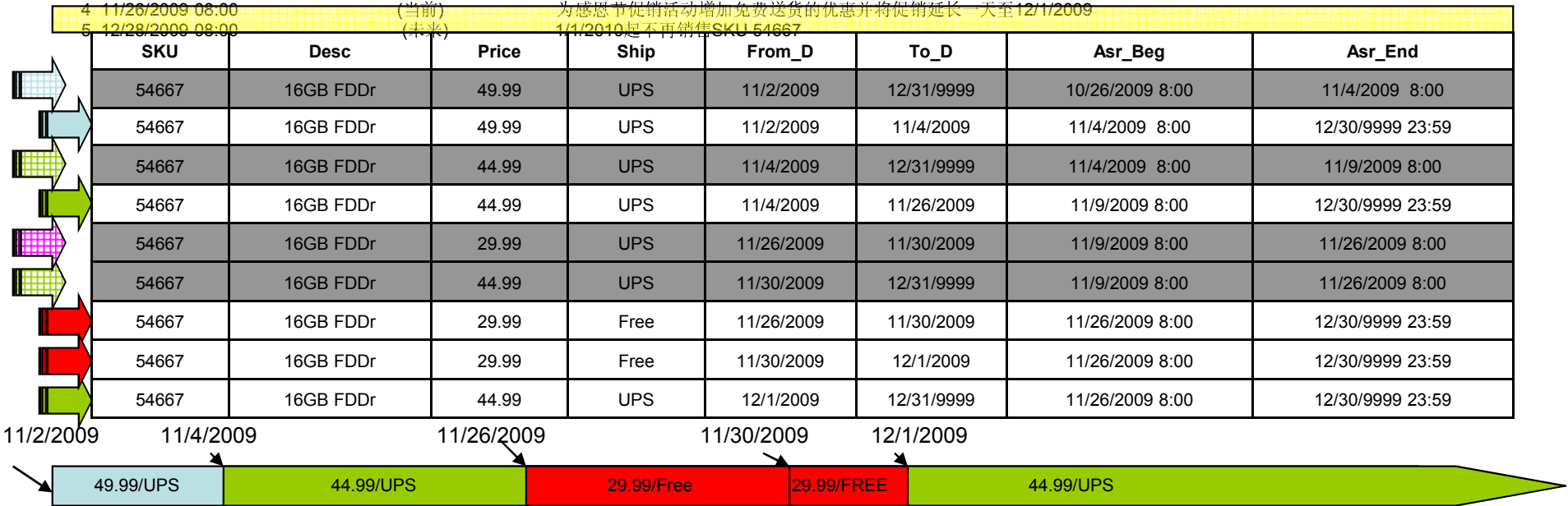
UPDATE Lstng FOR PORTION OF BUSINESS_TIME FROM DATE '11/26/2009' TO '12/1/2009'
 SET Price='29.99' Ship='Free' WHERE SKU='54667'

11/2/2009 11/4/2009 11/26/2009 11/30/2009 12/1/2009

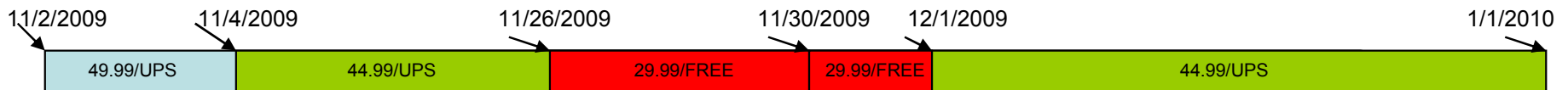


	SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
	54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
	54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/30/9999 23:59
	54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	11/9/2009 8:00
	54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009	11/9/2009 8:00	12/30/9999 23:59
	54667	16GB FDDr	29.99	UPS	11/26/2009	11/30/2009	11/9/2009 8:00	11/26/2009 8:00
	54667	16GB FDDr	44.99	UPS	11/30/2009	12/31/9999	11/9/2009 8:00	11/26/2009 8:00
	54667	16GB FDDr	29.99	Free	11/26/2009	11/30/2009	11/26/2009 8:00	12/30/9999 23:59
	54667	16GB FDDr	29.99	Free	11/30/2009	12/1/2009	11/26/2009 8:00	12/30/9999 23:59
	54667	16GB FDDr	44.99	UPS	12/1/2009	12/31/9999	11/26/2009 8:00	12/30/9999 23:59

- 1 10/26/2009 08:00 (未来) 从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99
- 2 11/4/2009 08:00 (当前) 将SKU 54667的价格下调为 \$44.99
- 3 11/9/2009 08:00 (未来) 安排从11/26/2009至11/30/2009以\$29.99的价格开展SKU 54667的感恩节促销
- 4 11/26/2009 08:00 (当前) 为感恩节促销活动增加免费送货的优惠并将促销延长一天至12/1/2009
- 5 12/28/2009 08:00 (未来) 11/2010起不再销售SKU 54667



DELETE FROM Lstng FOR PORTION OF BUSINESS_TIME FROM DATE '1/1/2010' TO '12/31/9999'
WHERE SKU='54667'

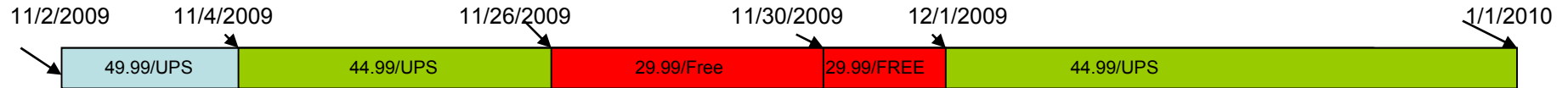


SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	11/9/2009 8:00
54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009	11/9/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	29.99	UPS	11/26/2009	11/30/2009	11/9/2009 8:00	11/26/2009 8:00
54667	16GB FDDr	44.99	UPS	11/30/2009	12/31/9999	11/9/2009 8:00	11/26/2009 8:00
54667	16GB FDDr	29.99	Free	11/26/2009	11/30/2009	11/26/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	29.99	Free	11/30/2009	12/1/2009	11/26/2009 8:00	12/30/9999 23:59
54667	16GB FDDr	44.99	UPS	12/1/2009	12/31/9999	11/26/2009 8:00	12/28/2009 8:00
54667	16GB FDDr	44.99	UPS	12/1/2009	11/1/2010	12/28/2009 8:00	12/30/9999 23:59

业务时间用例：

- 1 10/26/2009 08:00 (未来) 从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99
- 2 11/4/2009 08:00 (当前) 将SKU 54667的价格下调为 \$44.99
- 3 11/9/2009 08:00 (未来) 安排从11/26/2009至11/30/2009以\$29.99的价格开展SKU 54667的感恩节促销
- 4 11/26/2009 08:00 (当前) 为感恩节促销活动增加免费送货的优惠并将促销延长一天至12/1/2009
- 5 12/28/2009 08:00 (未来) 1/1/2010起不再销售SKU 54667

SKU	Desc	Price	Ship	From_D	To_D
54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009
54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009
54667	16GB FDDr	29.99	Free	11/26/2009	11/30/2009
54667	16GB FDDr	29.99	Free	11/30/2009	12/1/2009
54667	16GB FDDr	44.99	UPS	12/1/2009	1/1/2010



业务问题 1: 客户于 2009 年 12 月 26 日在一家 B&M 店铺要求退回 SKU 54667, 但没有收据。应该为客户支付多少退款?

SELECT MIN(PRICE) FROM Lstng FOR BUSINESS TIME FROM CURRENT DATE – 1 YEAR TO CURRENT DATE WHERE SKU='54667';

业务问题 2: 客户于 2010 年 1 月 25 日致电在线店铺, 表示她找到了 2009 年 11 月 26 日在线购买 SKU 54667 的收据。B&M 仅退给她 29.99 美元, 这位客户要求我们退还所收取的送货费, 并且表示我们的在线广告说感恩节促销期间免送货费。我们不会在线保留一个月以上的销售记录副本, 而且这位客户是一位高价值客户。那么这位客户是否确实支付了送货费?

无重叠的关键字

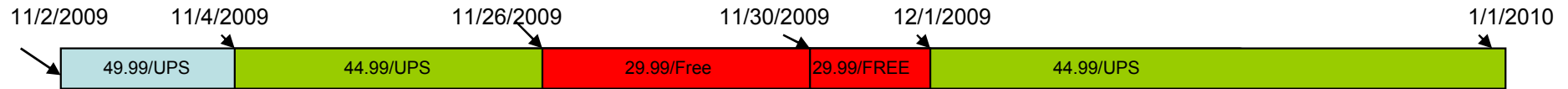
	SKU	Desc	Price	Ship	From_D	To_D
→	54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009
→	54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009
→	54667	16GB FDDr	29.99	Free	11/26/2009	11/30/2009
→	54667	16GB FDDr	29.99	Free	11/30/2009	12/1/2009
→	54667	16GB FDDr	44.99	UPS	12/1/2009	1/1/2010

Timeline diagram showing price and shipping changes for SKU 54667:

- 11/2/2009 - 11/4/2009: 49.99/UPS
- 11/4/2009 - 11/26/2009: 44.99/UPS
- 11/26/2009 - 11/30/2009: 29.99/Free
- 11/30/2009 - 12/1/2009: 29.99/FREE
- 12/1/2009 - 1/1/2010: 44.99/UPS

- 惟一约束
 - 在各业务时间点惟一
 - 例如，（SKU）就是一个无重叠的关键字
- 规范
 - 在DDL上
- 实施
 - 通过特别的 DB2 内部结构实现自动和高效的实施

SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	49.99	UPS	11/2/2009	12/31/9999	10/26/2009 8:00	11/4/2009 8:00
54667	16GB FDDr	49.99	UPS	11/2/2009	11/4/2009	11/4/2009 8:00	12/31/9999 23:59
54667	16GB FDDr	44.99	UPS	11/4/2009	12/31/9999	11/4/2009 8:00	11/9/2009 8:00
54667	16GB FDDr	44.99	UPS	11/4/2009	11/26/2009	11/9/2009 8:00	12/31/9999 23:59
54667	16GB FDDr	29.99	UPS	11/26/2009	11/30/2009	11/9/2009 8:00	11/26/2009 8:00
54667	16GB FDDr	44.99	UPS	11/30/2009	12/31/9999	11/9/2009 8:00	11/26/2009 8:00
54667	16GB FDDr	29.99	Free	11/26/2009	11/30/2009	11/26/2009 8:00	12/31/9999 23:59
54667	16GB FDDr	29.99	Free	11/30/2009	12/1/2009	11/26/2009 8:00	12/31/9999 23:59
54667	16GB FDDr	44.99	UPS	12/1/2009	12/31/9999	11/26/2009 8:00	12/28/2009 8:00
54667	16GB FDDr	44.99	UPS	12/1/2009	11/1/2010	12/28/2009 8:00	12/31/9999 23:59



- 1 10/26/2009 08:00 (未来) 从11/2/2009起将SKU 54667 16GB FDDr标为\$49.99
- 2 11/4/2009 08:00 (当前) 将SKU 54667的价格下调为 \$44.99
- 3 11/9/2009 08:00 (未来) 安排从11/26/2009至11/30/2009以\$29.99的价格开展SKU 54667的感恩节促销
- 4 11/26/2009 08:00 (当前) 为感恩节促销活动增加免费送货的优惠并将促销延长一天至12/1/2009
- 5 12/28/2009 08:00 (未来) 1/1/2010起不再销售SKU 54667

业务：客户于 2010 年 1 月 25 日致电在线店铺，表示她将 2009 年 11 月 26 日在线购买 SKU 54667 的收据交给了退给她 29.99 美元的 B&M 店铺，这位客户要求我们退还所收取的送货费，并且表示我们的在线广告说感恩节促销期间免送货费。一个月以上的销售收据副本保留在存档中，而且这位客户是一位高价值客户。我们能否确认客户是否支付了送货费？我们请她提供她的收据上打印的销售时间。她回答说是 2009 年 11 月 26 日上午 7 点。

```
SELECT * FROM Lstng FOR SYSTEM_TIME AS OF '11/26/2009 7AM'
FOR BUSINESS TIME AS OF '11/26/2009' WHERE SKU='54667';
```

SKU	Desc	Price	Ship	From_D	To_D	Asr_Beg	Asr_End
54667	16GB FDDr	29.99	UPS	11/26/2009	11/30/2009	11/9/2009 8:00	11/26/2009 8:00

生产力方面的节约

项目	用户应用解决方案	DB2 提供的解决方案
系统时间支持	每个表 2 个触发器	由 DB2 提供
业务时间支持	每个表 2 个存储过程	由 DB2 通过 SQL 语句提供
期间重叠检测	每个表 1 个触发器	各表的部分 PK 索引
代码总行数	650	13
SQL 语句总数	27	3
开发和测试所需时间	6 周	不到 1 小时

议程

- IBM Information Management
- 双时态数据库（BiTemporal Database）
 - 用例和业务场景
 - SQL 扩展
 - 示例
- 细粒度访问控制
 - 用例和业务场景
 - SQL 扩展
 - 示例

关于数据库安全性的忧虑

- 责任分离
 - DBADM 等数据库管理员可以访问敏感数据
 - 没有 SECADM 等专门权威来管理安全性策略
- 特权模型的粒度
 - 在数据库对象级授予特权
 - 难以保护对象内的个人和敏感信息
 - 无法轻松遵从数据保护法规，例如 HIPPA、GLBA
- 大量安全性逻辑使应用程序负担过重
 - 可被恶意用户绕过
 - 妨碍利用即席（ad-hoc）查询工具、报表生成工具的能力
 - 难以维护
- 各用户组的不同视图
 - 视图的更新可能无法正确反映安全性策略
 - 可被恶意用户绕过
 - 难以维护
- 安全性策略的演进
 - 影响应用程序中的安全性逻辑
 - 影响组织和视图数量

细粒度访问控制：行级和列级

- 易于实现
 - 通过 **SQL** 提供更高的灵活性
 - 安全性逻辑与应用程序逻辑分离
- 易于维护
 - 无需管理大量视图；不存在视图可更新性的问题
 - 无需给应用程序造成过重的负担
- 更严密的安全性
 - 数据库内以数据为中心
 - 视图或应用程序不存在可绕过的后门
 - 通过行权限和列屏蔽提供更细的粒度
 - 责任分离
 - 专门的 **SECADM** 权威
 - 无权威可免受控制，包括 **DBADM** 在内
 - 减少安全性策略的演变
- 用户友好
 - 无需记住各种视图或应用程序的名称
 - 允许使用即席查询工具、报表生成工具

何谓行级和列级访问控制？

■ 行筛选

- 谁能查看哪些行
- 适用于 SELECT、INSERT、UPDATE、DELETE、MERGE
- 行权限（新对象）

```
CREATE PERMISSION permission-name ON table-name  
FOR ROWS WHERE search-condition  
ENFORCED FOR ALL ACCESS;
```

■ 列屏蔽

- 谁能查看一列中的哪些值
- 适用于最外层子查询的输出
 - 对比较、分组、排序等无影响
- 列屏蔽（新对象）

```
CREATE MASK mask-name ON table-name  
FOR COLUMN column-name RETURN CASE-expression ENABLE;
```


何谓行级和列级访问控制？

■ 谁？

– **SESSION_USER**

- 进程的主要授权 ID

– **CURRENT SQLID**

- 进程的 SQL 授权 ID
- SET CURRENT SQLID = *string-constant*;

– **CURRENT USER**（新增特殊寄存器）

- 用于语句授权的授权 ID
- 确定何时绑定一条语句
- 无法设置
- 如果一个角色用于语句授权，则包含 NULL 值

– **CURRENT STATEMENT ROLE**（新增特殊寄存器）

- 用于语句授权的角色
- 确定何时绑定一条语句
- 无法设置
- 如果一个授权 ID 用于语句授权，则包含 NULL 值

何谓行级和列级访问控制？

- 如何？ ？ ？

- **VERIFY_GROUP_FOR_USER** (新增 BIF)

- 获取作为 SESSION_USER 中的值的授权 ID
 - 主要和辅助授权 ID
 - 如果其中任何授权 ID 处于参数列表中，则返回 1

```
WHERE  
  VERIFY_GROUP_FOR_USER (SESSION_USER, 'MGR', 'PAYROLL') = 1
```

- **VERIFY_ROLE_FOR_USER** (新增 BIF)

- 获取作为 SESSION_USER 中的值的角色
 - 主要授权 ID 的角色
 - 如果角色处于参数列表中，则返回 1

```
WHERE  
  VERIFY_ROLE_FOR_USER (SESSION_USER, 'MGR', 'PAYROLL') = 1
```

何谓行级和列级访问控制?

- 列屏蔽

```
CREATE MASK INCOME_COLUMN_MASK ON CUSTOMER
FOR COLUMN INCOME RETURN
CASE WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER, 'CSR') = 1)
      THEN CASE WHEN (INCOME > 150000) THEN 4
                WHEN (INCOME > 75000)  THEN 3
                WHEN (INCOME > 25000)  THEN 2
                ELSE 1
      END
      ELSE NULL
END
ENABLE;
```

何谓行级和列级访问控制？

■ 激活行级和列级访问控制

- 使行权限和列屏蔽在 DML 中生效
 - 所有行权限都使用“OR”连接，以筛选行
 - 所有列屏蔽都应用于屏蔽输出
- 如果没有用户定义的行权限，则阻止对表的所有访问

```
ALTER TABLE table-name  
  ACTIVATE ROW LEVEL ACCESS CONTROL  
  ACTIVATE COLUMN LEVEL ACCESS CONTROL;
```

■ 停用行级和列级访问控制

- 使行权限和列屏蔽在 DML 中无效
- 允许对表的所有访问

```
ALTER TABLE table-name  
  DEACTIVATE ROW LEVEL ACCESS CONTROL  
  DEACTIVATE COLUMN LEVEL ACCESS CONTROL;
```

银行场景

- 表: CUSTOMER

Account	Name	Phone	Income	Branch
1111-2222-3333-4444	Alice	111-1111	22,000	A
2222-3333-4444-5555	Bob	222-2222	71,000	B
3333-4444-5555-6666	Louis	333-3333	123,000	B
4444-5555-6666-7777	David	444-4444	172,000	C

- 表: CUSTOMER_CHOICE

Account	Phone_Choice
1111-2222-3333-4444	1
2222-3333-4444-5555	1
3333-4444-5555-6666	0
4444-5555-6666-7777	1

- 表: INTERNAL_INFO, 存储关于银行员工的信息

银行场景

- 第 1 步： < SECADM >
 - 确定出纳员的访问控制规则
 - 仅可通过其分行访问客户 (行)
 - 不允许查看 INCOME 列 (列)
 - 均使用辅助授权 ID TELLER (谁)
 - 为出纳员创建一个行权限

```
CREATE PERMISSION TELLER_ROW_ACCESS ON CUSTOMER
FOR ROWS WHERE

    VERIFY_GROUP_FOR_USER (SESSION_USER, 'TELLER') = 1
AND
    BRANCH = (SELECT HOME_BRANCH FROM INTERNAL_INFO
              WHERE EMP_ID = SESSION_USER)

ENFORCED FOR ALL ACCESS;
```

银行场景

- 第 2 步： < SECADM >
 - 确定客户服务代表的访问控制规则。
 - 可以访问银行的所有客户 (行)
 - 允许查看 INCOME 列的一般值(列)
 - 25000 表示收入为 25000 及 25000 以下
 - 75000
 - 150000
 - 999999 表示收入超过 150000
 - 均使用辅助授权 ID CSR (谁)
 - 为客户服务代表创建一个行权限

```
CREATE PERMISSION CSR_ROW_ACCESS ON CUSTOMER
FOR ROWS WHERE

    VERIFY_GROUP_FOR_USER (SESSION_USER, 'CSR') = 1
ENFORCED FOR ALL ACCESS;
```

银行场景

- 第 2 步：续.....
 - 在 **INCOME** 列上为出纳员和客户服务代表创建列屏蔽

```
CREATE MASK INCOME_COLUMN_MASK ON CUSTOMER
FOR COLUMN INCOME RETURN
CASE WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER, 'CSR') = 1)
    THEN CASE WHEN (INCOME > 150000) THEN 999999
              WHEN (INCOME > 75000) THEN 150000
              WHEN (INCOME > 25000) THEN 75000
              ELSE 25000
    END
ELSE NULL
END
ENABLE;
```


银行场景

- 第 3 步: < SECADM >
 - 确定同时作为电话销售人员的客户服务代表的访问控制规则
 - 仅可查看帐户号码的后四位数字 (列)
 - 允许在客户个人选择加入/选择退出的基础上查看客户电话号码 (列)
 - 均使用辅助授权 ID TELEMARKETING (谁)
 - 为电话销售人员创建列屏蔽

```
CREATE MASK ACCOUNT_COLUMN_MASK ON CUSTOMER
FOR COLUMN ACCOUNT RETURN

CASE WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER,'TELEMARKETING') = 1)
      THEN 'xxxx-xxxx-xxxx-' || SUBSTR(ACCOUNT,16,4)
      ELSE ACCOUNT
END
ENABLE;
```

银行场景

- 第 3 步：续.....

- 为电话销售人员创建列屏蔽

```
CREATE MASK PHONE_COL_MASK ON CUSTOMER
FOR COLUMN PHONE RETURN
CASE WHEN (VERIFY_GROUP_FOR_USER (SESSION_USER, 'TELEMARKETING' ) = 1)
        THEN CASE WHEN(EXISTS (SELECT 1 FROM CUSTOMER_CHOICE C
                                WHERE CUSTOMER.ACCOUNT = C.ACCOUNT AND
                                C.PHONE_CHOICE = 1)
                )
        THEN PHONE
        ELSE NULL
        END
    ELSE PHONE
    END
ENABLE;
```

银行场景

- 第 4 步: < SECADM >
 - 激活行级和列级访问控制

```
ALTER TABLE CUSTOMER
  ACTIVATE ROW LEVEL ACCESS CONTROL
  ACTIVATE COLUMN LEVEL ACCESS CONTROL;
COMMIT;
```

- DB2 中发生了什么?
 - 隐式创建了默认行权限，阻止对 CUSTOMER 表的所有访问
 - 所有引用 CUSTOMER 表的包和缓存语句均变为无效

银行场景

- 第 5 步： < 支行 B 的出纳员 Mary >

```
SELECT ACCOUNT, NAME, INCOME, PHONE FROM CUSTOMER;
```

结果：

ACCOUNT	NAME	INCOME	PHONE
2222-3333-4444-5555	Bob	?	222-2222
3333-4444-5555-6666	Louis	?	333-3333

银行场景

- 第 6 步： < 客户服务代表 Peter >

```
SELECT ACCOUNT, NAME, INCOME, PHONE FROM CUSTOMER;
```

结果：

ACCOUNT	NAME	INCOME	PHONE
1111-2222-3333-4444	Alice	25,000	111-1111
2222-3333-4444-5555	Bob	75,000	222-2222
3333-4444-5555-6666	Louis	150,000	333-3333
4444-5555-6666-7777	David	999,999	444-4444

银行场景

- 第 7 步： < 电话销售人员 Paul >

```
SELECT ACCOUNT, NAME, INCOME, PHONE FROM CUSTOMER;
```

结果：

ACCOUNT	NAME	INCOME	PHONE
xxxx-xxxx-xxxx-4444	Alice	25,000	111-1111
xxxx-xxxx-xxxx-5555	Bob	75,000	222-2222
xxxx-xxxx-xxxx-6666	Louis	150,000	?
xxxx-xxxx-xxxx-7777	David	999,999	444-4444



谢谢

Gene Fuhr, Ph.D.
fuhr@us.ibm.com

