INFO20003: Database Systems

Dr Renata Borovica-Gajic

Lecture 24

Overview, sample exam questions

Part II

Week 12

Query Processing –formulae

- Query Processing Cost Formulae on the LMS

RF selectivity proportion of data satisfy the condition

min max → provided in the system catelog

# of distinct value

① col = val $RF = \frac{1}{n\,keys}$

② col > val $RF = \frac{max - val}{max - min}$

③. val1 < col < val2 $RF = \frac{val2 - val2}{max - min}$

Query Processing 1/2

Consider relations *Employees*, *Orders* and *OrderDetails*. Imagine that relation Employees has 1,000 pages, relation Orders 5,000 pages, and relation OrderDetails 10,000 pages. Each page stores 100 tuples, and neither relation has any indexes built on it. Consider the following query:

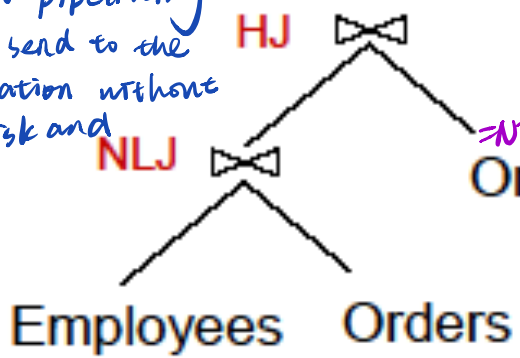
100 T/page

```
SELECT *
FROM Employees as E, Orders as O, OrderDetails as OD
WHERE E.empid = O.empid AND O.orderid = OD.orderid;
```

1000 pages 5000 pages 10000 pages

Compute the cost of the plan shown below. NLJ is a *Page-oriented* Nested Loops Join. Assume that *empid* is the candidate key of Employees, *orderid* is the candidate key of Orders, and 100 tuples of a resulting join between Employees and Orders can fit on one page.

Left deep plan allow pipelining (output is directly send to the input of next operation without writing back on disk and read again).



cost(NLJ) = NPages(E) + NTuple(E) × NPages(O)
= 1000 + 1000 × 5000
= 5001000

resulting size(E⋈O)
= NTuple(E) × NTuple(O) × RF
~~= 1000.00 × 5000.00 × 1/10000~~

= 5000 pages

cost(HJ) = 2 × 5000 + 3 × 10000
= 130000

cost = 5131000

Query Processing 2/2

Consider the query presented below. Does the following equivalence class hold? Yes/No and Why?

```
SELECT firstname, lastname
FROM Employees NATURAL JOIN Orders NATURAL JOIN OrderDetails
WHERE quantity > 5 AND freight < 100
```

No

$$\Pi_{\text{firstname, lastname}}(\sigma_{\text{quantity} > 5 \wedge \text{freight} < 100}(\text{Employees} \bowtie \text{Orders} \bowtie \text{OrderDetails}))$$

inner most execute first

$$\leftrightarrow \sigma_{\text{quantity} > 5 \wedge \text{freight} < 100}(\Pi_{\text{firstname, lastname}}(\text{Employees} \bowtie \text{Orders} \bowtie \text{OrderDetails}))$$

X

projection didn't preserve quantity, freight.

Normalization

The table shown below is part of an office inventory database. Identify the design problems and draw a revised table structure in 3rd Normal Form (3NF) that corrects those problems. For each step explicitly identify and discuss which normal form is violated.

Item ID is the candidate key for this table. Item ID determines Description, Quan, Cost/Unit and Dept, while Dept determines Dept Name and Dept Head.

1st one attribute has multiple values.
2nd forbid partial dependency
3rd forbid transitive dependency

(can simple remove it) derived attribute

Item ID	Description	Dept	Dept Name	Dept Head	Quan	Cost/Unit	Inventory Value
4011	5 ft desk	MK	Marketing	Jane Thompson	5	200	1000
4020	File cabinet	MK	Marketing	Jane Thompson	10	75	750
4005	Executive chair	MK	Marketing	Jane Thompson	5	100	500
4036	5 ft desk	ENG	Engineering	Ahmad Rashere	7	200	1400

itemID → description, Quan, Cost/Unit, Dept

Dept → DeptName, DeptHead.

1NF the relation is already in the 1NF

2NF the relation is already in the 2NF

Case of repeated group

eID	N	surname	phone number
1	John	Joe.	1111, 333, 888, 444

violate 1NF.

ID	Number

eid ID number

foreign key.

3NF.

Department (Dept, deptName, Dept Head)

Inventory (ItemID, description, Quan, cost/Unit, dept FK)

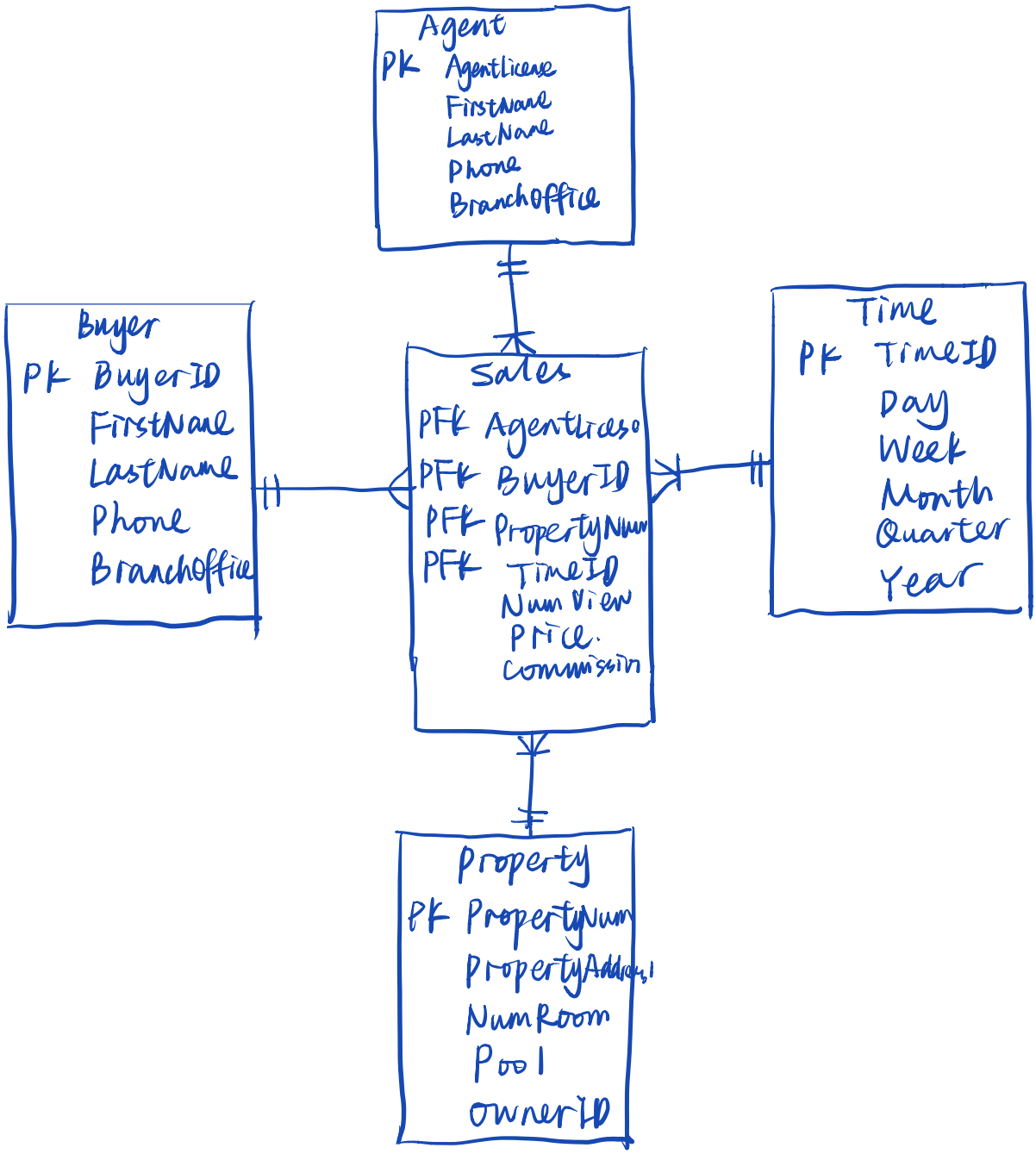
Data Warehouse

You are making a data warehouse for a **real estate** agency. The company wants to track information about the **selling** of their properties. Whenever a buyer comes into the office an agent takes that person to a number of properties and deals with the buyer. This warehouse keeps information about the **agents** (real estate license#, first name, last name, phone #, and branch office), **buyers** that come in (buyer id, first name, last name, phone #, max price), and **property** (property#, property address, number of rooms, pool, owner id). The information managers want to be able to find is **the number of times a property is viewed**, **sales price** and **commission**. Sales commission is additional compensation the agent receives for exceeding expectations. The information needs to be accessible **by agent**, **by buyer**, **by property** and **for different time** (day, week, month, quarter and year). Draw a star schema to support the design of this data warehouse.

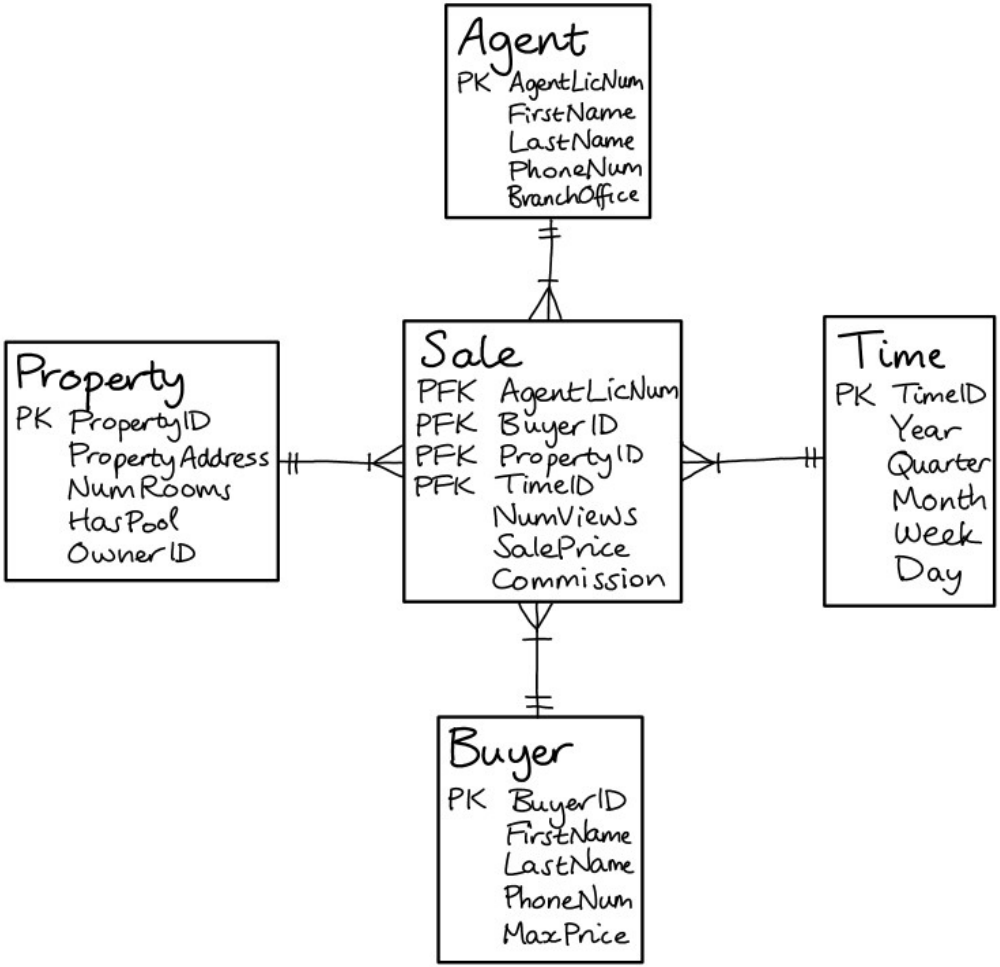
granularity

finest level.

4 dimension table



Data Warehouse - Solution



DW questions

- What is and how to identify business process
  - What the study is talking about (short couple words description)
  - E.g. “weather forecasting”, “real-estate sales”
- How to choose grain
  - Look at each dimension individually and see what is the finest needed level per that dimension (measurements are at the intersection of all finest)
  - E.g. if monitoring rainfall and we need to be able to report hourly, daily, weekly and monthly rainfall – we need to store *hourly* rainfall as a measurement. We can derive the value of courser granularities such as weekly from hourly measurements, but can’t do the opposite (from weekly get hourly)

Region
city state country
per city per hour

Database admin

Vine is a social media sharing service where users can host 6 second video clips within multiple categories (e.g. “Comedy”, “Science”, “Social”). Part of the database schema for the Vine service is given below.

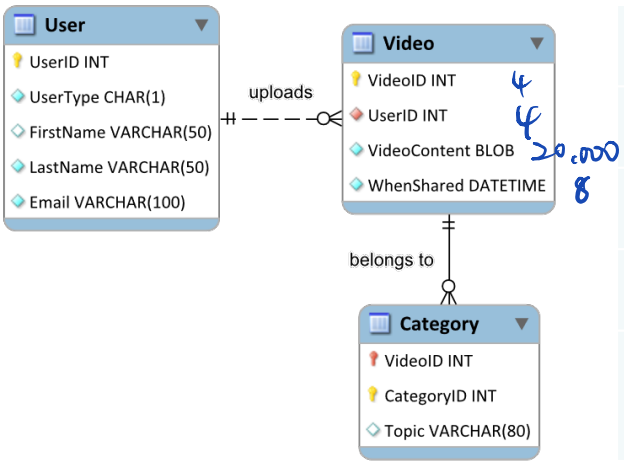
There are 15 different categories that users can share videos about and 1 million users to start with. A user posts 5 videos on average per month. Assume that the average storage requirement for the BLOB data type is 20,000 bytes.

Estimate the disk space requirements **only for the Video table** at go-live and after one month of operation.

moment for creation

estimate size ① average tuple size. 2001b

1) go live size(go live) = 0 2) size(one month) = tuple size × 1m × 5



Storage Requirement for different data types

Data Type	Storage requirements (bytes)
INT	4
BLOB	65,535 (Max)
DATETIME	8

= ... bytes → Kbytes → Mbytes → Gbytes.

Next lecture – no more ☺

THANK YOU!!!!!