Database Design Disasters

Richie Rump
Jorriss LLC
@Jorriss
www.jorriss.net

Who is this dude?

StatisticsParser.com

Why Database Design?

- Think of your database as your application foundation.
- The more time you spend on your foundation the better your application can be.
- If you get the foundation wrong you WILL have problems.

Why Database Design?

- Databases are highly complex systems.
- SQL Server has over twenty years of development.
- Learning SQL Server is like Alice's rabbit hole. It keeps going down.
- Leverage your DBAs experience and knowledge of SQL Server.

Why Database Design?

- Relational Databases are NOT going away!
- Reporting from NoSQL databases is difficult…for now.
- Right tool, right situation.

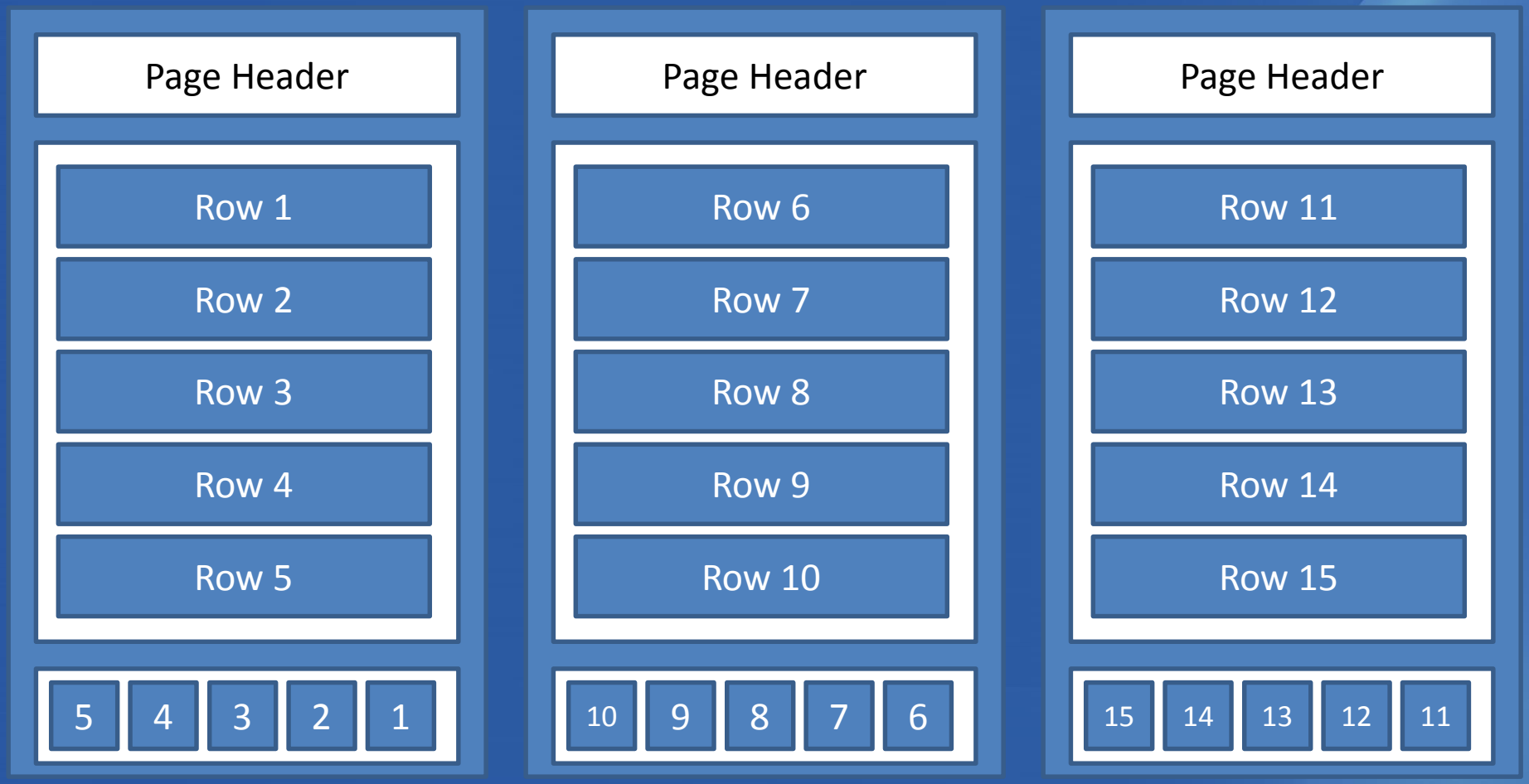
Current Toolset

- ORM tools like Hibernate, ActiveRecord and Entity Framework all can create databases.
- Out of the box schemas can have…opportunities.
- Agile frameworks and rapid iterations can leave DB design an afterthought.

Overview

- Wrong Types
- No Indexes on Foreign Keys
- Entity Attribute Value pattern
- Guids
- Surrogate Key / No Alternate Key

How SQL Server Stores Data



Poor Data Types

- Having the wrong data types can create long lasting issues.
- T-SQL can become challenging and inefficient.
- Understanding the differences between data types
- Better information, better decisions.

CHAR vs VARCHAR

- CHAR
  - Fixed length. Value will be padded with spaces.
  - Size *n* bytes.
  - 1 – 8000 characters.
- VARCHAR
  - Variable length.
  - The storage size is the actual length of the data entered + 2 bytes.
  - 1 – 8000 characters.

NCHAR vs NVARCHAR

- Holds UNICODE data.
- NCHAR
  - 1 – 4000 characters
  - Storage size twice *n* characters.
- NVARCHAR
  - 1 – 4000 character
  - Storage size twice *n* characters + 2 bytes.

NVARCHAR(MAX) / VARCHAR(MAX)

- MAX allows a column size of 2GB.
- Cannot create an index on MAX columns
- REPEAT: Cannot create an index on MAX columns
- NVARCHAR(MAX) is the default for a string when generated from Entity Framework.

DATETIME vs DATETIME2

- DATETIME
  - Accuracy rounded to .000, .003, or .007 second.
  - 8 Bytes
- DATETIME2 (*n*)
  - Accuracy to .0000001 second
  - 6 bytes for precisions less than 3;
  - 7 bytes for precisions 3 and 4.
  - All other precisions require 8 bytes.
- Demo

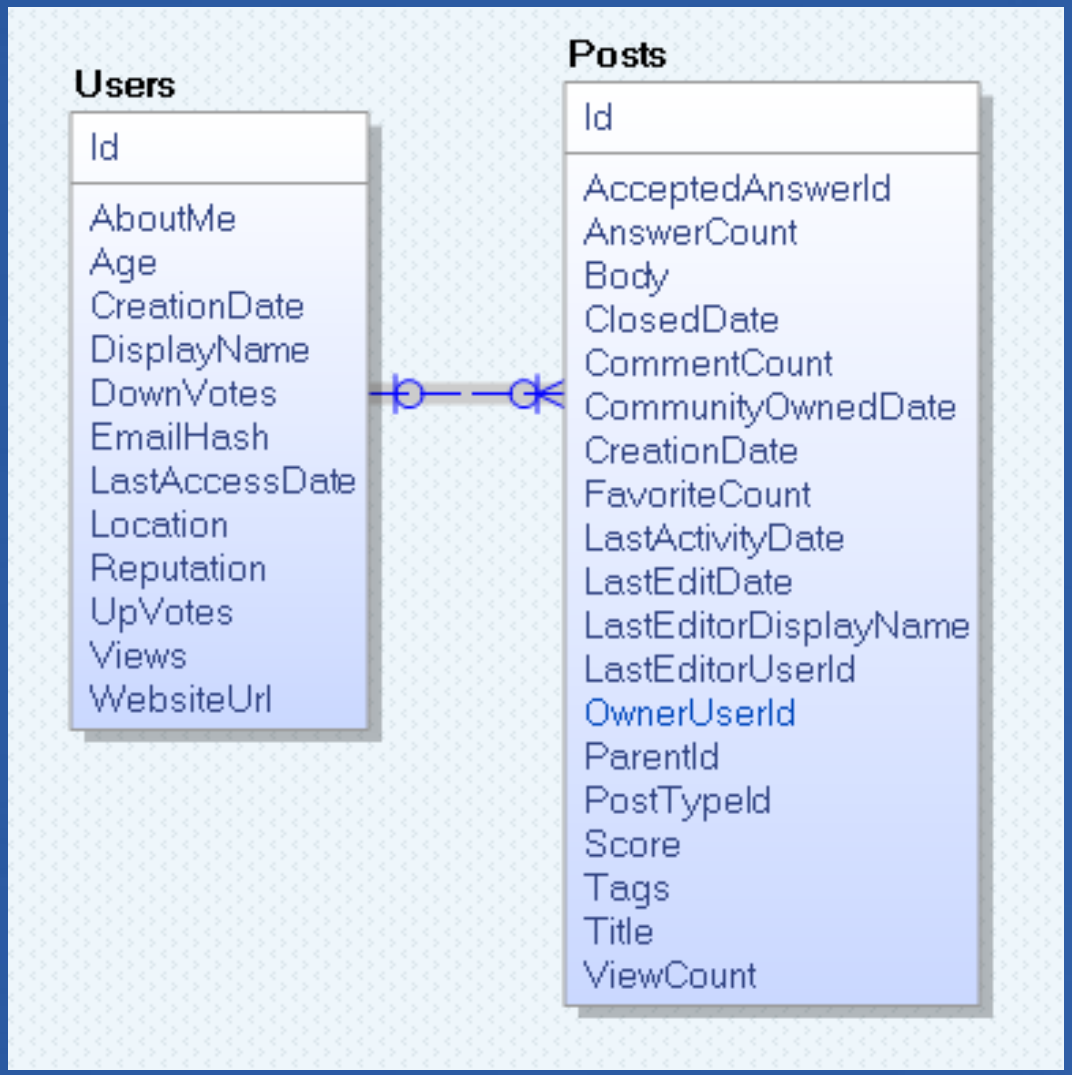
BIGINT vs INT

- BIGINT
  - -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
  - Size 8 bytes
- INT
  - -2,147,483,648 to 2,147,483,647
  - Size 4 bytes

No Indexes on Foreign Keys

- May slow queries with JOINs
- Will have a performance impact on DELETEs

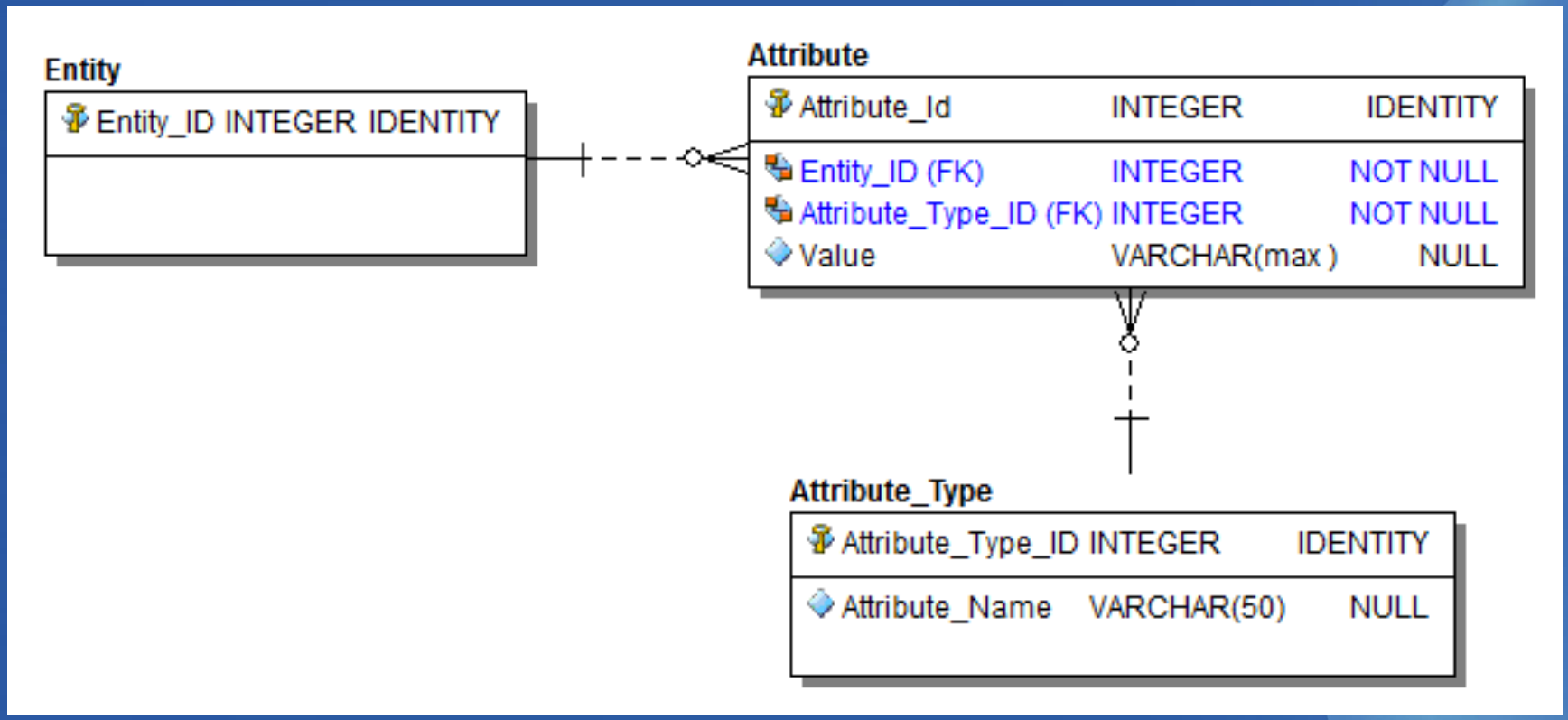
No Indexes on Foreign Keys



No Indexes on Foreign Keys

Demo

Entity-Attribute-Value



Entity-Attribute-Value

Person

Entity ID	Name
1	Cecil Phillip

Person_Attribute

Attribute ID	Entity ID	Attribute Type	Value
100	1	Phone Number	305-555-9607
101	1	Age	30
102	1	Birthdate	2/9/1983

Entity-Attribute-Value

- No Data Type Enforcement

Person

Entity ID	Name
1	Cecil Phillip

Person_Attribute

Attribute ID	Entity ID	Attribute Type	Value
100	1	Phone Number	305-555-9607
101	1	Age	501
102	1	Birthdate	Yes I have one

Entity-Attribute-Value

- No NOT NULL Enforcement

Person

Entity ID	Name
1	Cecil Phillip

Person_Attribute

Attribute ID	Entity ID	Attribute Type	Value
100	1	Phone Number	NULL
101	1	Age	30
102	1	Birthdate	2/9/1983

Entity-Attribute-Value

- Querying much harder
- Demo…

GUID Primary Keys

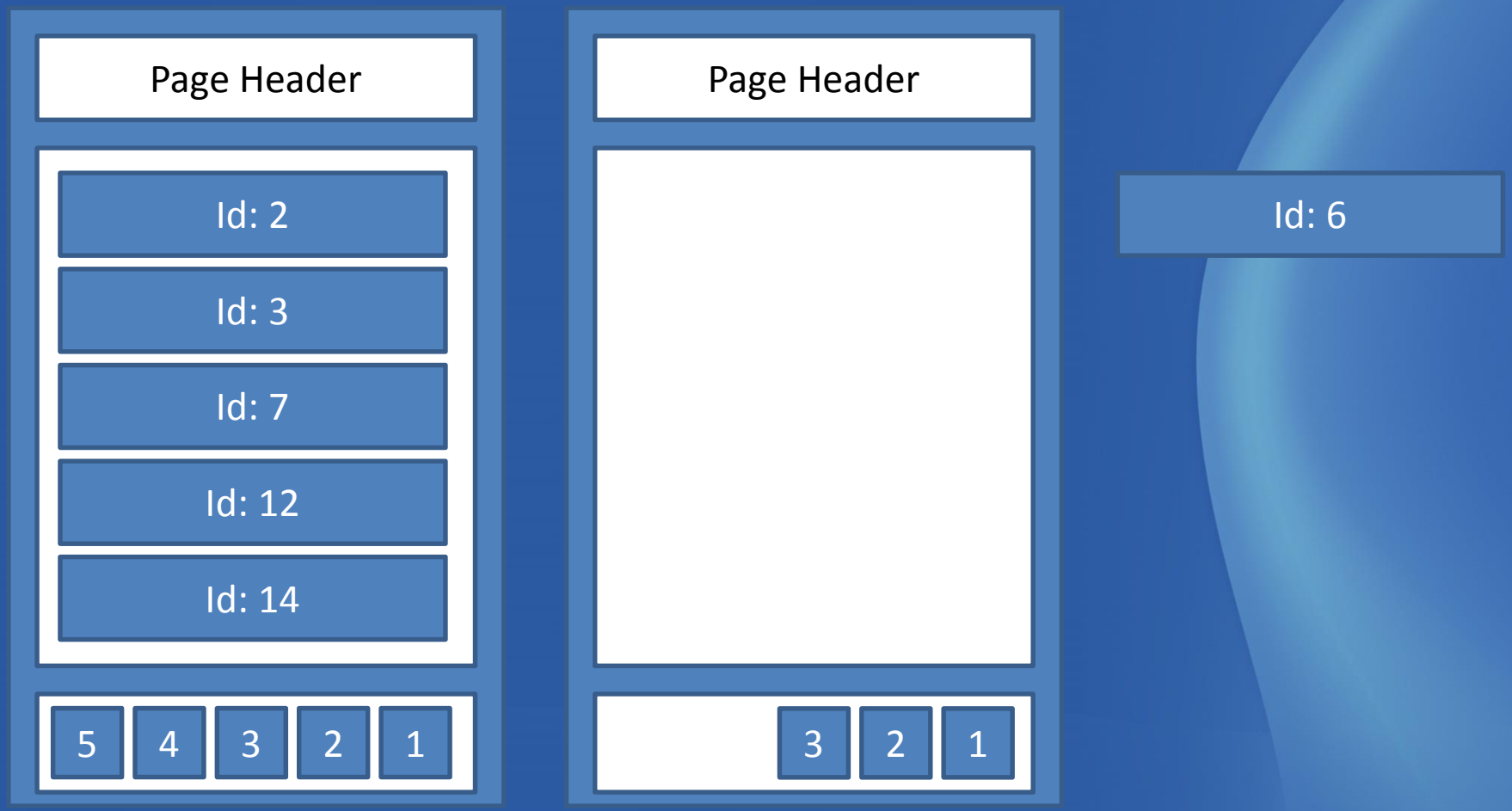
- You would think this wouldn’t be a problem…
- Some benefits
  - Portable
  - Id can be generated from the app
  - That’s about it.

GUID Primary Keys

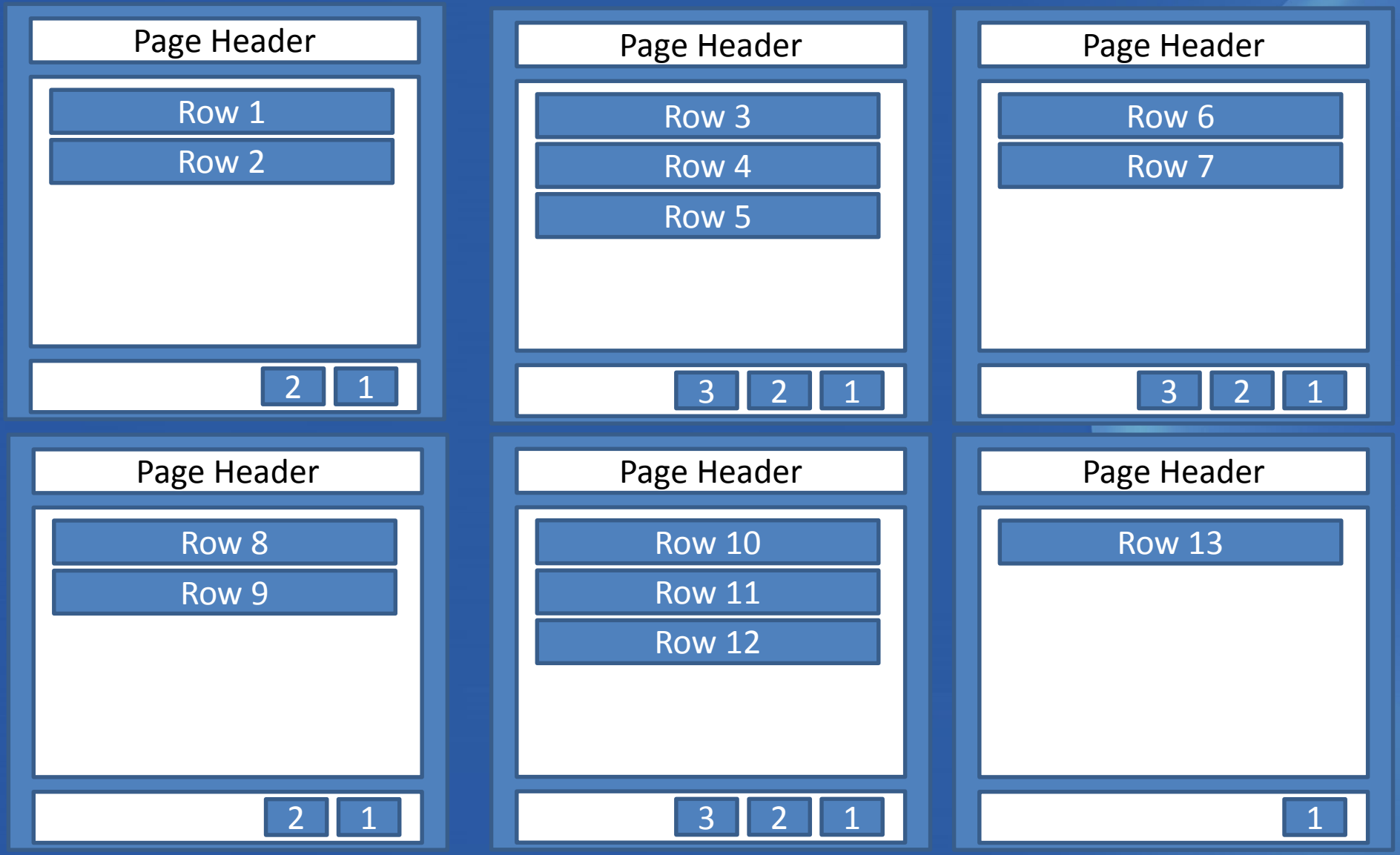
The problems:

- Size
  - Guids 16 bytes
  - Int 4 bytes
  - Disk and Memory
- Fragmentation

What Happens When a Page is Full



Fragmentation



GUID Primary Keys

A partial solution:

- NEWSEQUENTIALID()
  - Creates a sequential GUID.
  - Minimizes fragmentation
  - Still have the space issue.

Surrogate Key/No Alternate Key

- A surrogate key is a primary key automatically generated
- Surrogate keys are good thing.
- BUT when alternate key isn’t defined bad things can happen…bad things man, bad things.
- Demo…

Suggestions

- Align proper design with the requirements.
- Learn how SQL Server works
- Work more closely with the DB team.
- Document DB design decisions.

Reference

- **Data Model Resource Book vol 1-3**
  Len Silverston
- **Pro SQL Server 2012 Relational Database Design and Implementation**
  Louis Davidson, Jessica Moss

Thank You!!

Richie Rump

@Jorriss

http://jorriss.net

http://slideshare.net/jorriss

http://dotnetmiami.com