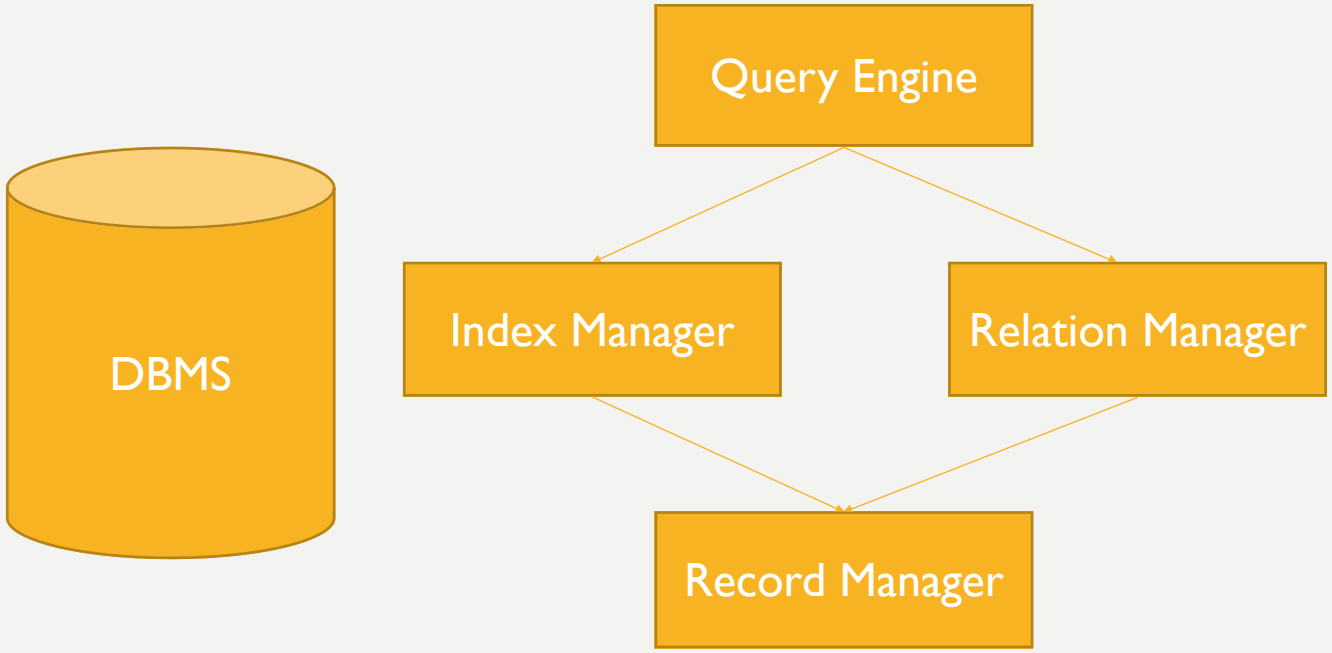
PROJECT INTRO

CHIYU CHENG

CYDB - ARCHITECTURE DESIGN



EXAMPLE - CREATE

- Create
  - CREATE TABLE Person (
    Age int,
    Name varchar(255);
    );
- Insert
- Search

CREATE – RELATION MANAGER

- System Catalog (Tables and Columns)
  - Tables

Table-id	Table-name	FileName
1	Tables	Tables.db
2	Columns	Columns.db

  - Columns

Table id	Column	Type	Length	Position
1	Table-id	Int	4	1
1	Table-name	VarChar	50	2
1	FileName	VarChar	50	3

AFTER CREATE TABLE PERSON

- CREATE TABLE Person (Age int, Name varchar(255););
- Tables

Table-id	Table-name	FileName
1	Tables	Tables.db
2	Columns	Columns.db
3	Person	Person.db

- Columns

Table id	Column	Type	Length	Position
3	Age	Int	4	1
3	Name	VarChar	50	2

WHY SYSTEM CATALOG?

- We can use it to generate an interpreter for one table to interpreter the binary file.
- Example interpreter for the Person table.
  - <

    <“Age”, “Int”, 1>,

    <“name”, “varChar”, 2>

    >

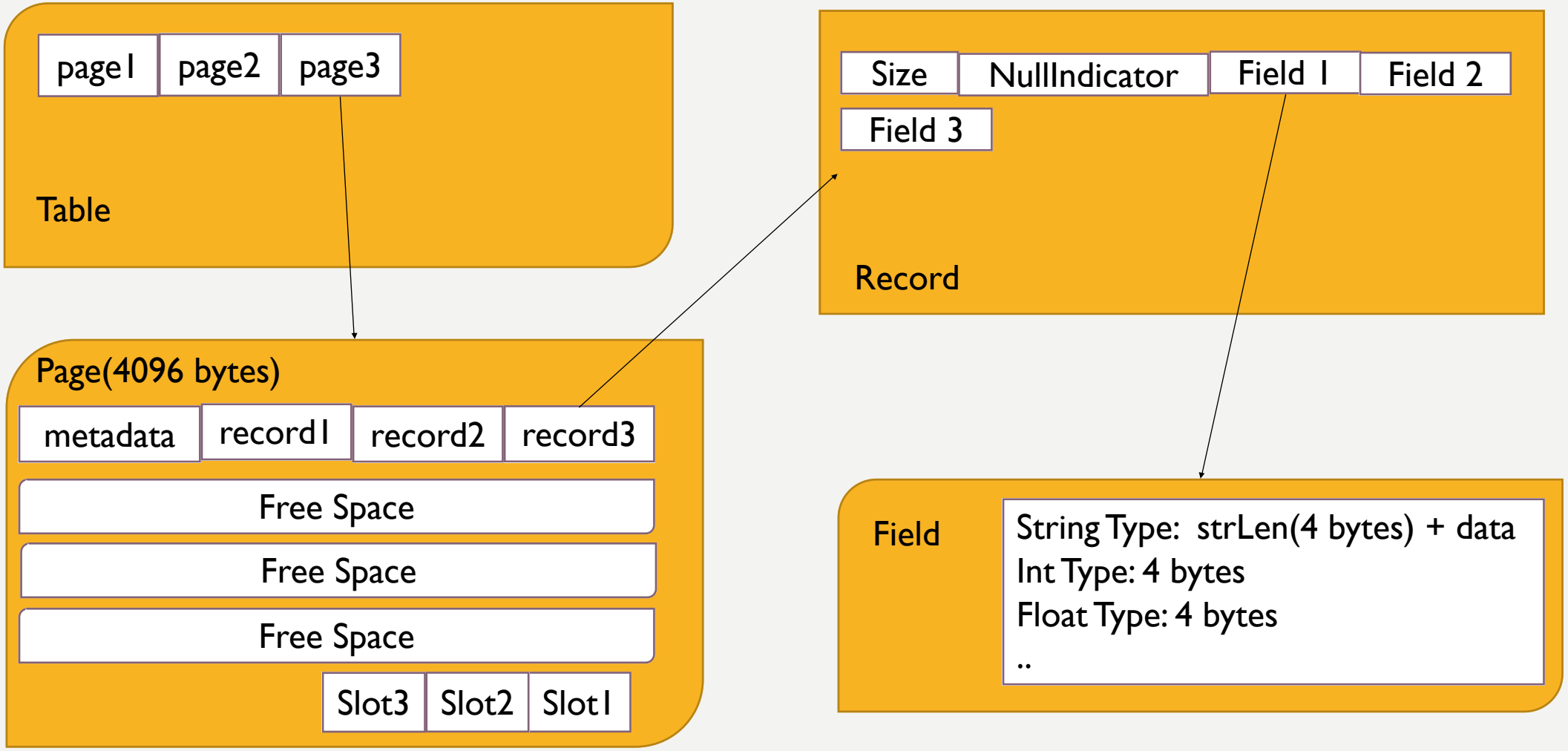
EXAMPLE - INSERT

- Create
  - CREATE TABLE Person (
    Age int,
    Name varchar(255);
    );
- Insert
  - INSERT INTO Person
    VALUES (*23,"Chiyu");*
- Search

RECORD BASED MANAGER DESIGN

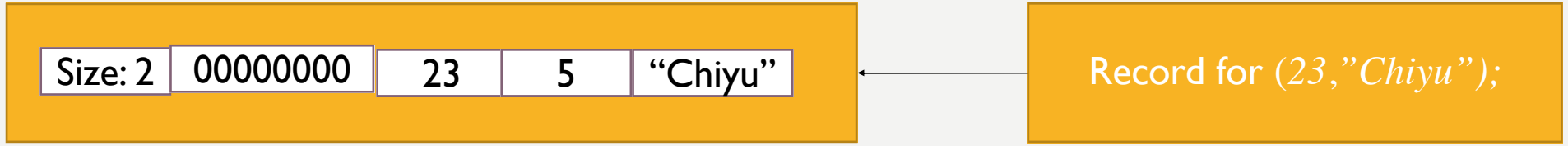
- Every table is a Binary file.
- The basic unit of a TABLE is PAGE(4096 bytes each).
- The basic unit of a PAGE is RECORD(No fixed size).
  - One row of each table.
- The basic unit of a RECORD is FIELD.
  - The data of each attribute for one column

RECORD BASED MANAGER DESIGN



INSERT

- INSERT INTO Person
  VALUES (*23,"Chiyu");*
- Use the generated interpreter to generate the record with binary format



Total size = 4 + 1 + 4 + 4 + 5 = 18 bytes

- Similar process with read/update/delete with RID(Record ID)
- Record Id = Page ID + Slot ID

EXAMPLE - SEARCH

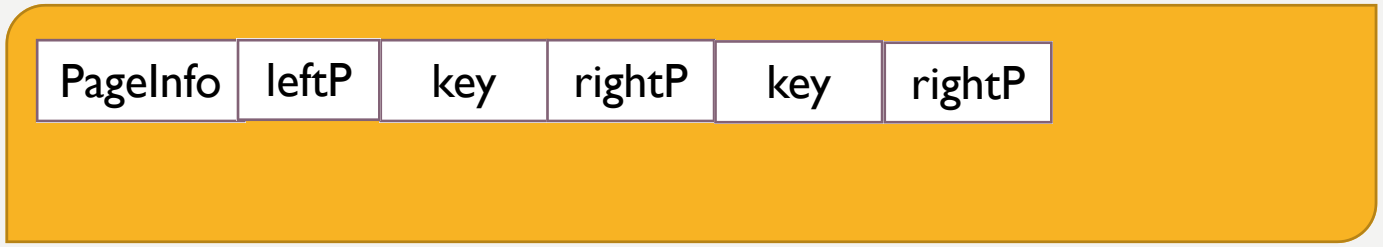
- Create
  - CREATE TABLE Person (
    Age int,
    Name varchar(255);
    );
- Insert
  - INSERT INTO Person
    VALUES (*23,"Chiyu");*
- Search
  - SELECT * FROM Person
    WHERE Age Between 20 AND 30;

INDEX MANAGER: B+ TREE INDEX

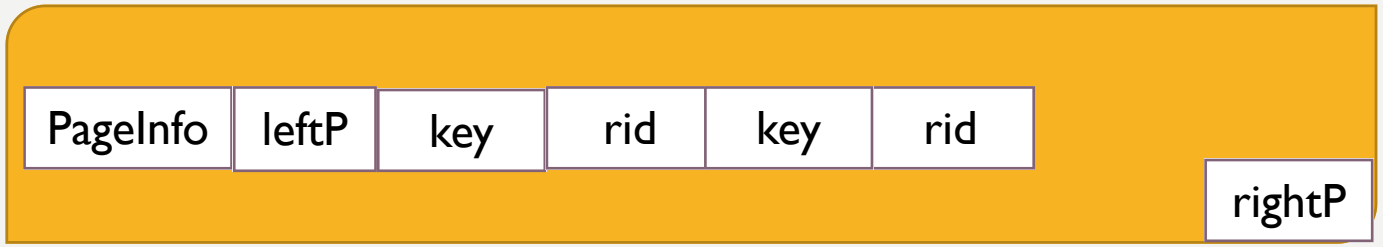
- Each index are build based on one attribute
- Each index is a binary file
- The basic unit of one index are
  - ixPage (4096 bytes)
  - ixLeafPage (4096 bytes)

IXPAGE AND IXLEAFPAGE

- IxPage – Index Page(4096 bytes) : the regular node in B+ tree



- IxLeafPage – Index Leaf Page(4096 bytes) : the leaf node in B+ tree



- PageInfo:
  - Number of Key, Data type, free space left

SEARCH

- SELECT * FROM Person
  WHERE Age Between 20 AND 30;
- Search through different index page to locate the right position in the leaf page. Keep loop to right until find all the key.
- Return the RID for each qualified key.
- Use the RID to search the record in the data table.

ADVANCED FUNCTION – QUERY ENGINE

- Join
- Nested Join
- Project – Select few attribute
- Min
- Max
- ….
- Use different iterators, indexScan, TableScan