DBMS

manage system _ function

‾ purpose

└ Sust data model ⟩

application & tools

Sql . & Tomcat (Apache)

Storage ∼ RAM (random - access

input — CPu — output   mory)。

↓

Arithmetic & logic unit   Second

(performane Calcolue)   Storage

unit

→ language for sustable programing
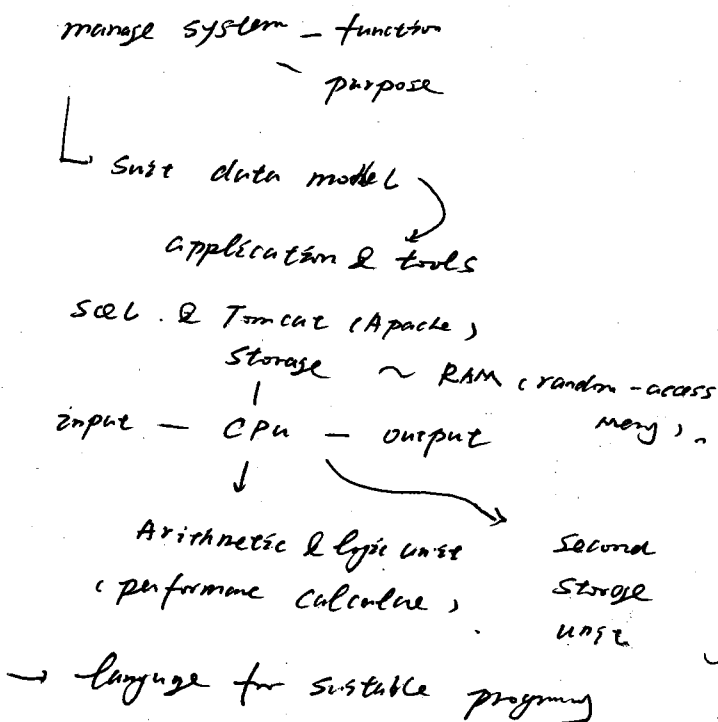
Machine . Assembly . high-level

— ARPA (advanced Research

Projects Agong · + DoD)

(TcP 6 . network acess - mere)

JSP . ASP. NET . Java- /Srcpt PHP DBMS ⟳

& . management DBMS

< Schema
  data

Schema ⟨==⟩ database

user ⟩ (inner connetion)

Database (sql 2005 - server)

└ schema

└ Table

(user Comtrol) (R / W)

Select current _ user

⇓

unit schema user .

shihang . table

① dbo   ② guest

③ SYS . ∼ ④ INFORMATION_

SCHEMA

MySol . 5.02 +

Create table ⇓

Create database = create schema

→ Application programmer .

└ end user

└ database administrator

(DBA)

① External   ⟩
② Conceptual   ⟩
③ Internal   ⟩

PS2 - data sub- language

< A . data definition L

B data manipulation . L

concept of database design

$\left\{ \begin{array}{l} \rightarrow \text{Conceptual design} \\ \rightarrow \text{logical design} \\ \rightarrow \text{physical design} \end{array} \right.$ $\left. \begin{array}{l} \\ \\ \end{array} \right\}$ Scope + boundaries
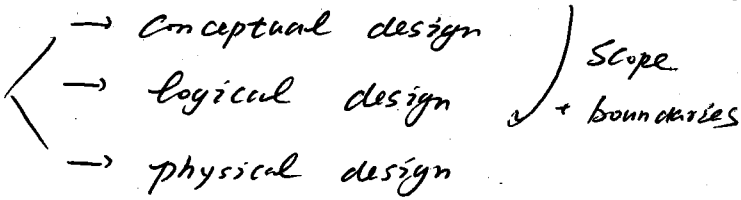
table name (primary key, Attribution 1~n, foreign key)

$\int$ primary key ~

~ Collection & analysis

model mapping

→ Entity & Relationship ~normalisation

ER (entity relationship) modelly

Entity & relationship & Attribute & Domain
(c & t) (c & t)

(Employee) ——N—— myer ——M—— (Project)
                        % the

primary - key

foreign - key   (unique) + Not Null)

(Foreign key (Id-s) References (Id-s)

Drop a foreign-key Id-s

→ Cardinality . (1:1) (1:n) (m:n)

A ——m—— hire ——n—— B

⇓

A ——< hire >—— B

ER . 2
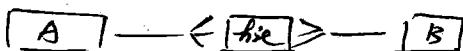
Normalization

1. 2. 3 ~ formal form

   └> Reduce data redundancies

     eliminate data anomalies

Integrity & redundancy

Anomalies ╱ update

         Addition

            ╲ Deletion .

1 NF → 2NF → 3NF.

make → easily Sort (entry-name)

Atomic ①atomic ⇐

       ② + primary key

repeating data must be in own table (2NF)

    ⇓

    Create new table   ↗

    ⇓

   fully functionally dependent ~ Fk.

    (no transitive dependences

        price

orderID, CustomerID, proID, price quantity. Total

      price × Quanty = total

'orderID, CusтомеID. ProdD, q....

   ² proID, quanty, price

~ database - sql

   create database (if not exist) data-name;

   Show databases;

   use database_name;

   Drop Database (if exists) database_name,

    ⎧ MyISAM × FK   ~select

    ⎪ InnoDB

    ⎨ Berkley DB (BDB)   ~ drop & import

    ⎪ MERGE

    ⎩ HEAP ×   InB/Text~   "set table-type = ~~"

    ⎧ set the_zne = "+00:00")

    ⎨ Insert into table-name value (~~

    ⎪ Select t1.

    ⎩ from table_name .

the_table (default) ~ ~

Create table -tall-name (

   employee - ID int (11) nrem

   ~   ~   ~

   ~   ~   ~

   PRImary key (employ -ID)

) TYpe = table_type

show tables;

describe tale - nare ;

Alter [Ignore] Table tclenare [~]

Alter tahe tale-nare ── add ~
── drop ~ /primryky
── change ~

Drop table (If exist) tale-nare ;

Truncate tale tale_nare ;

Select * from tale-nare ;

where (or)ID = 5 ; ── Conty in ~ (範围)
── Colmn = A prcal = B,

Select distinct * from tale-nare ;

+ ordered by (Disc) // gorp by

+ limit 5, 10 ;

+ Inner x = T ; 内联接

Left - join / right . 左/右 联接

SUM. AVG. MAX. MIN

Insert into table-nare

Select * From ~ where Cong = x" ;

Set (~ ~) were condition

Delete From tale-nare

where condi ~

Yarchar ~ decinal ~ date

INT ── tiy 3 int 10.
       smah 5
       medium 8   biy 20

Advance . sql

CREATE procedure nare ( )
Begin
select ~ ~ ~
End,

< call nare ( ) ; >

Decler tch-~ INT defo.
Set tar ~ = 10 ;
↓ IF . case
while loop,
Repen loop.

substring (str. pos)
((str From pos For ley)s

Select srbsy ("mysol siny"-p

ConCAT (str. sr₁)

update ~ sr

Basc - adme
sql 4.5

```html
<html> (xmlns = "~")
    <head>
        <title> ~ <title>
    </head>
<body>
</body>
</html>
```

Element
```html
<h1> ~ <h6>
<P> <strong> </P>
<a href = "http: ~~"> ~ </a>
<img src = "~ .jpg" height = "~" width = "~"/>
<br />

& copy  ©

<del> ✗✗ </del>
<sub>  ~² </sub>
```

```html
<ul>
    <li>
    <li>              unordered - list .
</ul>
<ol>
    <li>
    <li>              order - list
</ol>
```

table ✱   br ·

```html
<table border = "1" width = "40%">
    <caption> Band </caption>
    <thead>
        <tr>
            <th> </th>        "header"
            <th> </th>
        </tr>
    </thead>
    <tfoot>                             "footer"
        <tr>
            <th> </th>
            <th> </th>
        </tr>
    </tfoot>
    <tbody>
        <tr> <th></th>       "body"
        <tr> <th> </th> .
    </tbody>
</table>
```

```html
<hr>
```

```html
<th rowspan = "2">

</th>

<th colspan = "4" valign = "top">
```

html ✓

```
< form method = "post"
    action = " (cgi-bz/final">

  <input type = "hidden" name = "subject"
*        value = " Feedback Form" />

    <p> <label> name                    <input type = "password"
        <input type = "name" type="text"    / = "checkbix"  >
            size = "25" maxlength="30"/>      ☑ ~ ☑
          </lable>                         / = " radio"  ⊙
    </P>
                                         <textureq name=" comment"
    <P>                                      rows ="4" cls "10">
        <input type="submit" value="sub"/>
        <input type=" reset" value="reset"/>   </textures>.
    </P>

</form>
```

```
< select name = ' ~ ' >
    <option selected = "seleited"> sh </option>

        <option> ~ </option>    下拉菜单

</select>
```

```
    <h id = "features" > ~ </h>
<p>   <a href = "# bny" > go </a> <p>

                                     </a>
<p>  <a href = "# features" >go to feature> </p>

        <h id = "bny" > go go </h>
```

html⁺

数 学 作 业 纸

```
<map id="picture">
  <area href="form.html"
     shape = "rect" / "poly" ~
  Coords = "x. y. z...."
/#   ey: circle :  x. y. r.
       polygon : x₁y₁, x₂Y₂ "~
       rectangle : "x. Y₁, x₂r," ~
   alt = "value" />              #/
</map>

<img ... usemap = "#picture">
```

Element ⁺   <em><strong>

Frameset ~ element

```
<frameset col-"10.">
    <framename = "Left frame" src. ~.html/>
    <frame name = "main" src. ~.html>
   <no frame >
       Lbody>
       <body x>
    </no frame>
< / frameset >
```

CSS.

① inline styles

```
<P style = "front-size: 20pt;
  color = #000ff"> OK </P>
```

② embedded style

```
<head>
    <style type = "text/css">
  em    { background-color: ~;
            color: white }
  P      {              }
  h₁     {              }
  . special { color. ~ }
    </style>
```

⇒ `<h₁ class = "special"> </h₁>`
　`<P class = "special"> </P>`

③ Conflicting (同上)

```
    a. nodec { ~ }
    a: hower {    }
    li em   {    }
    ul      {    }

<li> ~ <em> X </em>

<a href = "~" class = nodec">
```

④ link external style

```
<link rel = "stylesheet"
    type = "text/css"
    href = "styles.css" />
```

Style.css [css]
(file) → [html]

```
<img src = ~ style = "position
  : absolute / left ~ ;
top : x Px ; left : x Px ;
z-index : n "> 
```

span CSS的中明后直接 ✓

`<span class = "sub"> ~ </span>`

```
. sub { position. relative fron-size: ~}
span { color: red ; }
```

```
{ Background
body
    - image : url (logo.png)
    - position : right
    - repeat : no-repi:
    - attchment ★ fixed. }
```
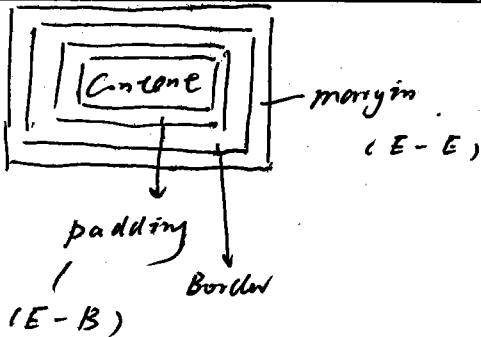
div style. " ~ ~ ~ "
↓↓
优先 > external   css !

Content

— margin
　(E - E)

↓ padding
↓ Border

(E - B)

inset　ouset

<div style ="border-style: solid"> </div>

double　grove

~ .note { front·size : PPx }

↳ <p class="note"> ^PPx^ ~~ </P>

优先级