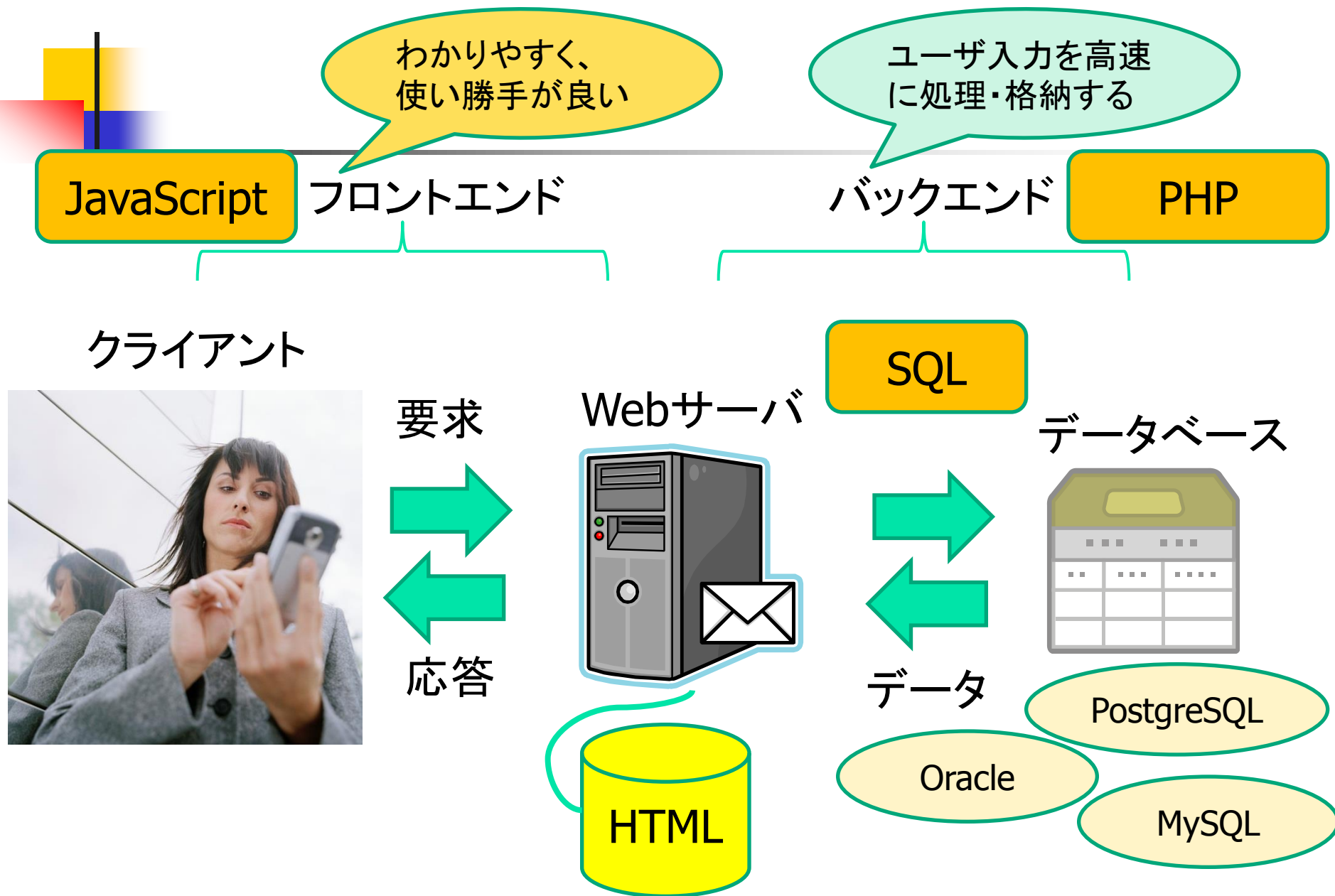


RDB入門 => PHPでアクセス

Webシステムの構成



リレーショナルデータベースのテーブル(表)

フィールド(列/column)



社員番号	氏名	入社年月日	部署コード
100	小滝 美子	2003-04-01	B001
101	波田 優子	2003-09-04	B002
104	石井 浩之	2003-04-01	B001

レコード(行/row)

主キーと外部キー

外部キー

社員番号	氏名	入社年月日	部署コード
100	小滝 美子	2003-04-01	B001
101	波田 優子	2003-09-04	B002
104	石井 浩之	2003-04-01	B001

主キー

参照

部署コード	部署名
B001	システム開発部
B002	総務部



RDB(リレーショナルデータベース)

- データベース => データの集まり
- データベース管理システム
 - => リレーショナルデータベース => SQL
 - テーブル(表) => データの格納
 - フィールド(列) => 情報を分類
 - レコード(行) => フィールドの集まり
 - 主キー => レコードを一意に識別できる
 - 外部キー => ほかのテーブルと関連付けられる



リレーショナルデータベース (MYSQL)演習



MySQL設定ファイル確認

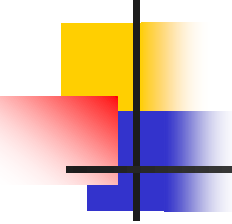
- # vi /etc/my.cnf or vi /etc/mysql/my.cnf
で、以下の追加。
[mysqld]
default-character-set=utf8 (5.5より前)
character_set_server=utf8 (5.5以降)
skip-character-set-client-handshake
default-storage-engine=innnoDB
innodb_file_per_table
[client]([mysql])
default-character-set=utf8
[mysqldump]
default-character-set=utf8
- MySQL再起動
/etc/init.d/mysqld restart



管理者ユーザ (ROOT) での接続

```
$ mysql -u root -p
```

Enter password: (パスワード入力)



データベース作成

管理者ユーザで、接続後、

```
mysql> create database testphp;
```

赤字はデータベース名、任意に指定。

以下でデータベースが作成されていることを確認

```
mysql> show databases;
```



ユーザ作成

管理者ユーザで、接続後、

```
mysql> grant all on testphp.* to  
test@localhost identified by 'pass';
```

赤字はデータベース名、青字はユーザ名、
黄字はパスワード任意。

以下でユーザが作成されていること確認

```
mysql> select user from mysql.user;
```



一般ユーザでの接続

- ユーザ名 => test
- パスワード => pass
- データベース名 => testphp

```
$ mysql -u test -p testphp  
Enter password: pass
```



MySQLから切断

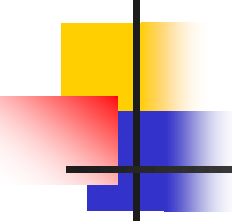
(切断)

>exit



実習用データ作成スクリプトの 実行

- `$ mysql -u test -p testphp < create_tbl_mysql.sql`
Enter password: **pass**



データベースの切替

> use データベース名;

(注) 以下の様にデータベースを指定せずにMySQLに
接続した場合に必要な

```
$ mysql -u test -p
```

```
Enter password: pass
```

```
> use testphp
```



テーブル一覧の表示

>show tables;



employees表から全列を表示

```
>select * from employees;
```




employees表からename(従業員名)を
表示

```
>select ename from employees;
```



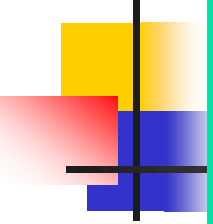
employees表からdeptno(部門番号)と
ename(従業員名)を表示

```
>select deptno,ename  
->from employees;
```



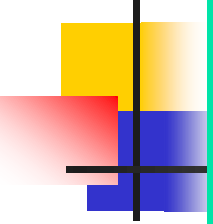
employees表からjob(職種)の種類を
表示

```
>select distinct job  
->from employees;
```



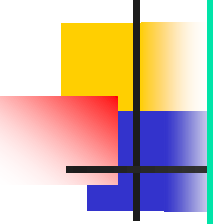
employees表から、job(職種)が「事務」の従業員を選択し、従業員名(ename)、職種(job)およびhiredate(入社日)を表示

```
>select ename, job, hiredate  
->from employees  
->where job = '事務';
```



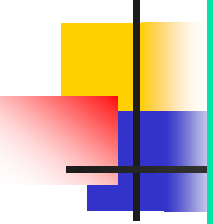
employees表から、給与(sal)が
300000以上の従業員を選択し、従
業員名(ename)および給与(sal)を
表示

```
>select ename, sal  
->from employees  
->where sal >= 300000;
```



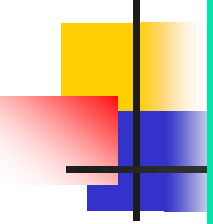
employees表から、入社日(hiredate)が'2000-01-01'から'2000-12-31'の従業員を選択し、従業員名(ename)および入社日(hiredate)を表示

```
>select ename, hiredate  
->from employees  
->where hiredate  
->between '2000-1-1' and '2000-12-31';
```



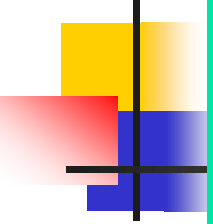
employees表から、吉田さん、山田さん、
佐々木さんを選択し、従業員番号(empno)、
従業員名(ename)、職種(JOb)および上司の
従業員番号(mgr)を表示

```
>select empno, ename, job, mgr  
->from employees  
->where ename  
->in ('吉田', '山田', '佐々木');
```



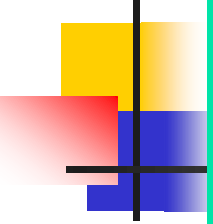
employees表から、従業員名(ename)
の2文字目が' 田'の従業員を選択し、
従業員名(ename)と職種(job)を表示

```
>select ename, job  
->from employees  
->where ename like '_田%';
```

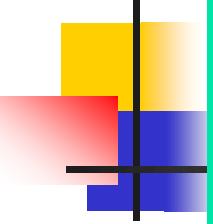
employees表から、上司の従業員番号(mgr)がNULLの従業員を選択して、従業員名(ename)および上司の従業員番号(mgr)を表示

```
>select ename, mgr  
->from employees  
->where mgr is null;
```



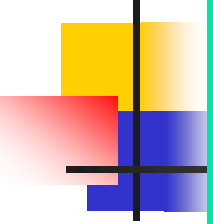
employees表から、給与(sal)が250000以下でかつ職種(job)が'事務'の従業員を選択して、従業員番号(employeesno)、従業員名(ename)、職種(job)、給与(sal)を表示

```
>select ename, job, sal  
->from employees  
->where sal <= 250000  
->and job = '事務';
```



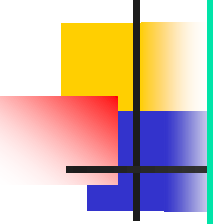
employees表から、職種(job)が'社長'、
'営業'以外の従業員を選択して、従業員
番号(empno)、従業員名(ename)、職
種(job)を表示

```
>select empno, ename, job  
->from employees  
->where job  
->not in ('社長', '営業');
```



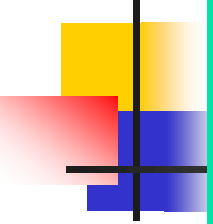
departments表のすべての列を表示する。この時、deptno(部門番号)の降順にソート

```
>select * from departments  
->order by deptno desc;
```



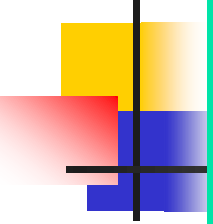
employees表から給料(sal)が200000
以下の従業員の従業員名(ename)、職
種(job)および給与(sal)を表示して、給
与(sal)の昇順にソート

```
>select ename, job, sal  
->from employees  
->where sal <= 200000  
->order by sal;
```



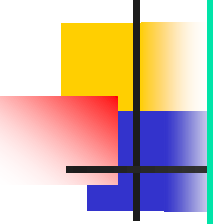
employees表から従業員番号(empno)、
従業員名(ename)、給与(sal) * 12 (列
別名 年収)を表示して、年収の降順に
ソート

```
>select empno, ename, sal*12 年収  
->from employees  
->order by 年収 desc;
```



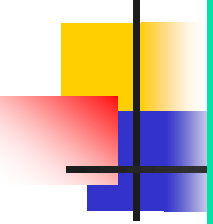
employees表から、従業員名(ename)、給与(sal)、歩合給(comm)、(給与+歩合給)*12を表示しなさい。ただし、歩合給がNULL値の場合はnullif関数を用いてゼロに置き換えて計算する

```
>select ename, sal, comm,  
->(sal + ifnull(comm, 0))*12 年収  
-> from employees;
```



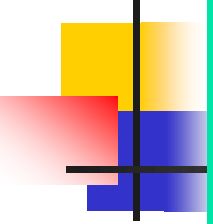
employees表から、従業員数及び従業員の給与について、その平均、最大値、最小値及び合計を表示

```
>select count(*), avg(sal), max(sal),  
->min(sal), sum(sal)  
-> from employees;
```

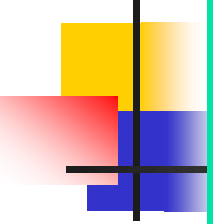
employees表から、部門番号ごとの給与の平均を表示しなさい。
この時、平均の昇順でソートする

- >select deptno, avg(sal) 平均
- >from employees
- >group by deptno
- >order by 平均;



部門番号ごとの給与の最高額を求めて、その中で最高額が300000を超える部門番号及び給与の最高額を表示

```
>select deptno, max(sal) 最高額  
->from employees  
->group by deptno  
->having 最高額 > 300000;
```



部門番号ごとの給与の平均を求めて、平均が300000を超える部門番号及び給与の平均額を表示

```
>select deptno, avg(sal) 平均  
->from employees  
->group by deptno  
->having 平均 > 300000;
```

employees表とdepartments表 の結合

employees表

empno
ename
job
mgr
hiredate
sal
comm
deptno

departments表

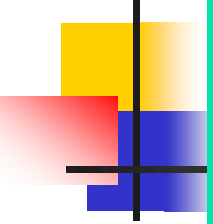
deptno
dname
loc

on employees.deptno =
departments.deptno



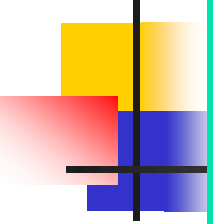
employees表及びdepartments表
から従業員番号(empno)、従業員
名(ename)、部門番号(deptno)、部
門のある場所(loc)を表示

```
>select empno, ename, e.deptno, loc  
->from employees e join departments d  
->on e.deptno = d.deptno;
```



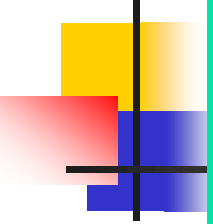
employees表及びdepartments表
から従業員番号(empno)、従業員
名(ename)、部門番号(deptno)、部
門のある場所(loc)を表示

>select empno, ename, deptno, loc
->from employees natural join departments;



employees表及びdepartments表から
従業員名(ename)が'鈴木'の従業員番
号(employeesno)、従業員名(ename)、
部門番号(deptno)、部門のある場所
(loc)を表示

```
>select empno, ename, e.deptno, loc  
->from employees e join departments d  
->on e.deptno = d.deptno  
->where ename='鈴木';
```



employees表及びdepartments表から
従業員名(ename)が'鈴木'の従業員番
号(employeesno)、従業員名(ename)、
部門番号(departmentsno)、部門のあ
る場所(loc)を表示

- >select empno, ename, deptno, loc
- >from employees natural join departments
- >where ename='鈴木';

customer表、ord表およびitem表の結合

customers表

custno
cname
address
phone
credit_rating

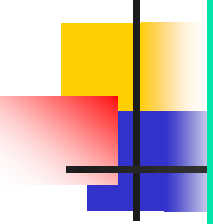
orders表

ordno
custno
date_ordered
date_shipped
salesman_no
payment_type

ord_details表

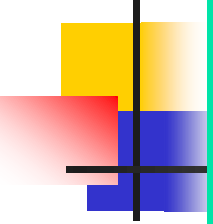
ordno
prodno
quantity

on customers.custno = orders.custno and
orders.ordno = order_details.ordno



customers表、orders表及びorder_details表から顧客名(name)が'横浜商店'の顧客名(cname)、注文番号(ordno)、製品番号(prodno)、注文日(date_ordered)、数量(quantity)を表示

```
>select cname, o.ordno, prodno,  
->date_ordered, quantity  
->from customers c join orders o  
->on c.custno = o.custno  
->join ord_details d  
->on o.ordno = d.ordno  
->where cname = '横浜商店';
```



customers表、orders表及びorder_details表から顧客名(name)が'横浜商店'の顧客名(cname)、注文番号(ordno)、製品番号(prodno)、注文日(date_ordered)、数量(quantity)を表示

```
>select cname, ordno, prodno,  
->date_ordered, quantity  
->from customers natural join orders  
->natural join ord_details  
->where cname = '横浜商店';
```

employees表とsalgrade表の結合

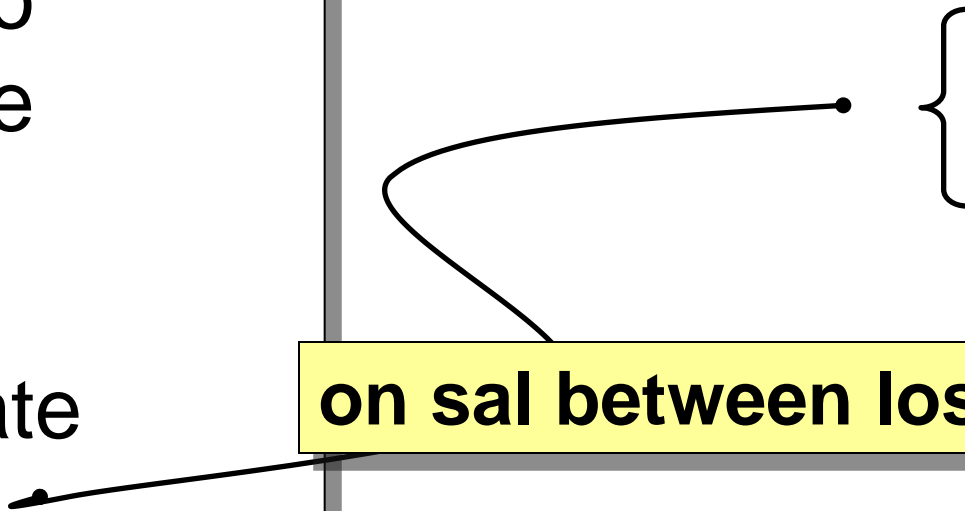
employees表

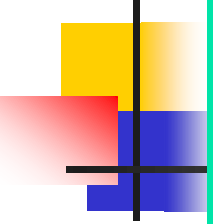
empno
ename
job
mgr
hiredate
sal
comm
deptno

salgrades表

grade
losal
hisal

on sal between losal and hisal





employees表、salgrade表(給与等級表)から従業員名(ename)、給与(sal)、下限給与(losal)、上限給与(hisal)及び給与等級(grade)を表示しなさい。
この時、給与が下限給与と上限給与の範囲に含まれるかどうかでemployees表とsalgrade表を関連付けする

- >select ename, sal, losal, hisal, grade
- >from employees join salgrades
- >on sal between losal and hisal;

employees表の自己結合

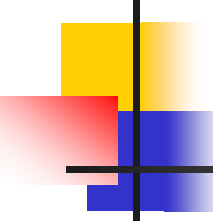
employees(e)表

empno
ename
job
mgr
hiredate
sal
comm
deptno

on e.mgr = m.empno

employees(m)表

empno
ename
job
mgr
hiredate
sal
comm
deptno



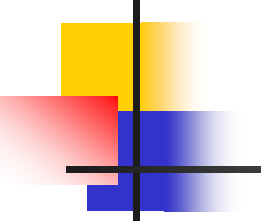
employees表を自己結合し、従業員番号(empno)、従業員名(ename)、職種(job)、上司の従業員番号(mgr)、上司の名前(ename)を表示する。この時、mgrの昇順にソートする

```
mysql> select e.empno, e.ename, e.job, e.mgr, m.ename  
-> from employees e left outer join employees m  
-> on e.mgr = m.empno  
-> order by e.mgr;
```



トランザクション制御

- オートコミットの確認
mysql> select @@autocommit;
- オートコミットの有効化
mysql> set autocommit=1;
- オートコミットの無効化
mysql> set autocommit=0;
- ロールバック
mysql> rollback;
- コミット
mysql> commit;

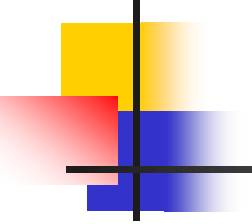


departments表へ
右表の値で行を追加し、正しく追加されたことを確認

列名	値
deptno	50
dname	製造
loc	沖縄

>insert into departments
->values(50, '製造', '沖縄');

>select * from departments;



departments表
へ右表の値で行
を追加し、正しく
追加されたこと
を確認しなさい。

列名	値
deptm entsno	60
dname	法務
loc	NULL

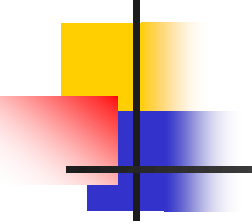
>insert into departments
->values(60, '法務', null);

>select * from departments;



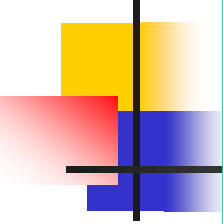
次の例題の準備

```
>create table employees2 as  
->select * from employees  
->where 0 = 1;
```



employees表から、職種(job)が'部長'のデータを
employees2表に挿入し、正しく追加されたことを
確認。

```
>insert into employees2  
->select *  
->from employees  
->where job = '部長';
```



employees表で、従業員番号(empno)が1011の従業員の部門番号(deptno)を40に変更。正しく変更されたことを確認。

>select empno, ename, deptno from employees


-> where empno=1011;

>update employees set deptno = 40

->where empno=1011;

>select empno, ename, deptno from employees

-> where empno=1011;

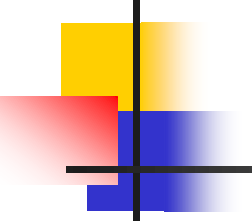


departments表からdname(部門名)
'財務'の行を削除。
また正しく削除されたことを確認。

>select * from departments;

>delete from departments where dname='法務';


>select * from departments;



deptno、dname、
locという3つの
列を持つ
departments2表
を作成し、確認。

列名	データ型
deptno	integer
dname	varchar(15)
loc	varchar(15)

```
>create table departments2(  
->deptno integer,  
->dname varchar(15),  
->loc varchar(15));  
  
>desc departments2;
```

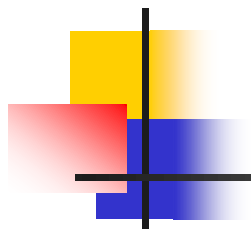


departments2表を削除し、削除されていることを確認

```
>show tables;
```

```
>drop table departments2;
```

```
>show tables;
```

PHPから接続



MDB2(pear)のインストール

- pearのインストール
 - # aptitude install php-pear (Debian系)
 - # yum install php-pear (Redhat系)
- 本体
 - # pear install MDB2
- ドライバ(MySQL)
 - #pear install MDB2_Driver_mysql



日本語対応

基本方針として、すべてUTF-8に統一する

- データベースの作成

「utf8_general_ci」を選択

```
mysql> create database {$DBname} default  
charset=utf8
```

- phpファイル

文字コードとして、「utf-8」を指定

「<meta charset="UTF-8">」記述し、文字コードを明示する

- MDB2のDSN

```
"mysql://{$User}:{$DBpass}@{$host}/{$DBname}?char  
set=utf8"
```

と、データベースの文字コードを指定



データベースとユーザ作成

- データベースとユーザを作成するため「root」ユーザでmysqlサーバに接続
\$ mysql -u root -p
Enter password: (パスワード入力)
- データベース作成
mysql> create database restaurant default charset = utf8;
- 上記のデータベースに接続するユーザ作成
mysql> grant all on restaurant.* to penguin@localhost identified by 'top^hot';
- 一旦、切断し、上記で作成したデータベースに作成したユーザで接続する
\$ mysql -u penguin -p restaurant
Enter password: (パスワード入力)