Default Layout Read table and column comments as mouse-over tooltips.

DbSchema 18-Jan-2019

employee
- employee_id #
- first_name t
- last_name t
- address t
- city t
- country c
- postal_code #
- nationality t
- mobile_number #
- alternate_contact_number #
- educational_background t
- employee_dob d
- designation_id #
- work_experience #
- experience_in t
- employee_dept_id #
- branch_id #
- desk_id t
- payroll_id #
- no_of_paid_leaves #
- employee_manager_id #
- employee_status b
- annual_bonus_amount #
- date_of_bonous_released d
- bonus_approver_id #
- employee_hr_id #
- grade_id #

company
- company_id #
- branch_id #
- city t
- country c
- state c
- street_number_name t
- postal_code #
- company_name t

desk_allocation
- desk_id t
- branch_id #
- status t

external_company_contract
- id #
- company_name t
- company_location t
- contract_duration #
- number_of_consultants #
- contract_amount #
- contract_with #

employee_attendance
- id #
- employee_id #
- mark_in_time d
- mark_out_time d
- employee_type_id #
- attendance_status t
- working_hrs #
- leave_type_id #
- calender_date d

leave_type
- id #
- type_of_leave t
- leave_description t
- no_of_leaves #

public_holiday
- id #
- date_of_holiday d
- occasion t
- company_state t

employee_type_mapping
- emp_type_id #
- description t
- id #
- consulting_comapny_id #
- contract_duration t
- designation_id #
- employee_id #

designation
- designation_id #
- designation_name t

aspirants
- aspirant_id #
- fisrt_name t
- last_name t
- job_id #
- work_experience #
- status c
- comments t
- job_location t
- hr_id #
- date_of_application d
- educational_background t
- employee_id #

job_openings
- job_id #
- job_position t
- job_description t
- is_active b
- no_of_employee_required #
- required_work_experience t
- skill_set t

leave_request
- id #
- employee_id #
- from_date d
- to_date d
- leave_type_id #
- comments t
- leave_approval_status b

employee_type
- emp_type_id #
- employee_type_name t
- emp_type_description t

external_company_contract_history
- id #
- company_name t
- company_location t
- contract_duration #
- number_of_consultants #
- contract_amount #
- contract_with #
- history_id #

department
- dept_id #
- dept_name t
- employee_id_hod #
- team_building_budget #

category
- id #
- categoty_name t

employee_reimbursement
- id #
- employee_id #
- reimbursement_amount #
- reimbursement_description t
- reimbursement_date d
- category_id #
- approval_status c

salary_increment
- id #
- grade t
- %increment #

asset_inventory
- asset_id #
- asset_name t
- asset_description t
- asset_insurance_id t
- asset_expiry_date d

ticket
- id #
- assigned_to_employee_id #
- raised_by_employee_id #
- ticket_description #
- ticket_status b
- ticket_tat d
- ticket_approval_status b
- approval_by t

payroll
- payroll_id #
- dept_id #
- salary_per_hr #

asset_liability
- id #
- asset_id #
- liable_employee_id #
- issued_date d
- returned_date d
- approval_status b

employee_history
- employee_id #
- first_name t
- last_name t
- address t
- city t
- country c
- postal_code #
- mobile_number #
- alternate_contact_number #
- educational_background t
- designation_id #
- work_experience #
- experience_in t
- employee_dept_id #
- branch_id #
- nationality t
- desk_id t
- payroll_id #
- no_of_paid_leaves #
- employee_manager_id #
- employee_status b
- id #
- payslip t
- employee_hr_id #

employee_salary_history
- id #
- payslip t
- increment_history #
- grade t
- increment_year d
- employee_id #

asset_inventory_history
- asset_id #
- asset_name t
- asset_description t
- aseet_insurance_id t
- liable_employee_id #
- returned_date d
- assigned_date d
- hr_approver_id #

employee_id
branch_id
leave_type_id
desk_id
designation_id
hr_id
job_id
employee_id_hod
employee_dept_id
grade_id
category_id
raised_by_employee_id
asset_id
assigned_to_employee_id
liable_employee_id
dept_id
payroll_id
comapny_id
contract_with
designation_id
emp_type_id
employee_id

Table aspirants for a particular job opening, if internal company employee applies then employee Id column will be filled. for a fresh outsider applicant, employee ID will be blank.			
*Pk	aspirant_id	integer	
Idx	fisrt_name	varchar(100)	
Idx	last_name	varchar(100)	
	job_id	bigserial	
	work_experience	integer	
	status	char(5)	
	comments	text	
	job_location	varchar(25)	
	hr_id	bigserial	
Idx	date_of_application	date	
	educational_background	text	
	employee_id	bigserial	
Indexes			
Pk	pk_aspirants_aspirant_id	aspirant_id	
	idx_aspirants_fisrt_name	fisrt_name	
	idx_aspirants_aspirant_id	aspirant_id	
	idx_aspirants_last_name	last_name	
	idx_aspirants_date_of_application	date_of_application	
Foreign Keys			
	fk_aspirants_employee (hr_id) ref employee (employee_id)		
	fk_aspirants_job_openings (job_id) ref job_openings (job_id)		
	fk_aspirants_employee_id (employee_id) ref employee (employee_id)		

Table asset_inventory this table will track the whole assets which company currently have in total.			
*Pk	asset_id	bigserial	
Idx	asset_name	varchar(50)	
	asset_description	varchar(100)	
	asset_insurance_id	varchar(100)	
	asset_expiry_date	date	
Indexes			
Pk	pk_employee_eqipment_id	asset_id	
	idx_asset_inventory_asset_id	asset_id	
	idx_asset_inventory_asset_name	asset_name	

Table asset_inventory_history			
*Pk	asset_id	bigserial	
Idx	asset_name	varchar(50)	
	asset_description	varchar(100)	
	aseet_insurance_id	varchar(100)	
	liable_employee_id	bigserial	
	returned_date	date	

Table asset_inventory_history			
	assigned_date	date	
	hr_approver_id	bigserial	
Indexes			
Pk	pk_employee_eqipment_id_0	asset_id	
	idx_asset_inventory_history_asset_id	asset_id	
	idx_asset_inventory_history_asset_name	asset_name	

Table asset_liability this table will track the issuing of asset to a particulatr employee.			
*Pk	id	bigserial	
	asset_id	bigserial	
Idx	liable_employee_id	bigserial	
	issued_date	timestamp	
	returned_date	timestamp	
Idx	approval_status	bool	
Indexes			
Pk	pk_asset_liability_id	id	
	idx_asset_liability_liable_employee_id	liable_employee_id	
	idx_asset_liability_approval_status	approval_status	
Foreign Keys			
	fk_asset_id (asset_id) ref asset_inventory (asset_id)		
	fk_asset_liable_employee_id (liable_employee_id) ref employee (employee_id)		

Table category			
*Pk	id	bigserial	
	categoty_name	varchar(50)	
Indexes			
Pk	pk_ticekt_category_id	id	

Table company single organization can have different sub companies like Volkswagen owns companies like Audi,BentleyPorsche etc.each subcompany can have its branches at different locations. to make a table entries unique, branch_id will be composite key of company_id+Branch_id.			
*Pk	company_id	bigserial	
Unq	branch_id	bigserial	
	city	varchar(25)	
	country	char(25)	
	state	char(50)	
	street_number_name	varchar(50)	
	postal_code	integer	
	company_name	varchar(50)	
Indexes			
Pk	pk_company_company_id	company_id	
	compositekey_comanyid_branchid	company_id, branch_id	

Table company single organization can have different sub companies like Volkswagen owns companies like Audi,BentleyPorsche etc.each subcompany can have its branches at different locations. to make a table entries unique, branch_id will be composite key of company_id+Branch_id.			
Unq	unq_company_branch_id	branch_id	

Table department			
*Pk	dept_id	bigserial	
Idx	dept_name	varchar(50)	
	employee_id_hod	bigserial	
	team_building_budget	float8	
Indexes			
Pk	pk_department_dept_id	dept_id	
	idx_department_dept_id	dept_id	
	idx_department_dept_id_0	dept_id	
	idx_department_dept_name	dept_name	
Foreign Keys			
	fk_department_person (employee_id_hod) ref employee (employee_id)		

Table designation			
*Pk	designation_id	bigserial	
	designation_name	varchar(25)	
Indexes			
Pk	pk_designation_designation_id	designation_id	
	idx_designation_designation_id	designation_id	

Table desk_allocation			
*Pk	desk_id	varchar	
	branch_id	bigserial	
Idx	status	varchar(10)	
Indexes			
Pk	pk_desk_allocation_desk_id	desk_id	
	idx_desk_allocation_status	status	
Foreign Keys			
	fk_desk_allocation_company (branch_id) ref company (branch_id)		

Table employee 1. employee organziational hirarchy will be maintained by self looping to this table using employee_manager_Id and employee_Hr_Id.2. annual bonus will be updated in at the end of financial year(or may be if company decide to give bonus half yearly).3. on separation, the employee_status will be deactivated. employee data will be soft-deleted to keep the history.			
*Pk	employee_id	bigserial	
	first_name	varchar(50)	
Idx	last_name	varchar(50)	
	address	varchar(150)	
	city	varchar(25)	
	country	char(25)	
	postal_code	integer	
	nationality	varchar(25)	

Table employee 1. employee organziational hirarchy will be maintained by self looping to this table using employee_manager_Id and employee_Hr_Id.2. annual bonus will be updated in at the end of financial year(or may be if company decide to give bonus half yearly).3. on separation, the employee_status will be deactivated. employee data will be soft-deleted to keep the history.			
	mobile_number	integer	
	alternate_contact_number	integer	
	educational_background	varchar(25)	
	employee_dob	date	
Idx	designation_id	integer	
	work_experience	bigserial	
	experience_in	text	
Idx	employee_dept_id	bigserial	
Idx	branch_id	bigserial	
	desk_id	varchar	
	payroll_id	bigserial	
	no_of_paid_leaves	bigserial	
	employee_manager_id	bigserial	
	employee_status	bool	
	annual_bonus_amount	float8	
	date_of_bonous_released	date	
	bonus_approver_id	bigserial	
	employee_hr_id	bigserial	
Idx	grade_id	bigserial	
Indexes			
Pk	pk_person_id	employee_id	
	idx_employee_employee_id	employee_id	
	idx_employee_employee_id_0	employee_id	
	idx_employee_last_name	last_name	
	idx_employee_designation_id	designation_id	
	idx_employee_employee_dept_id	employee_dept_id	
	idx_employee_branch_id	branch_id	
	idx_employee_grade_id	grade_id	
Foreign Keys			
	fk_person_department (employee_dept_id) ref department (dept_id)		
	fk_person_company (branch_id) ref company (branch_id)		
	fk_employee_desk_allocation (desk_id) ref desk_allocation (desk_id)		
	fk_employee_designation (designation_id) ref designation (designation_id)		
	fk_employee_payroll (payroll_id) ref payroll (payroll_id)		
	fk_employee_salary_increment (grade_id) ref salary_increment (id)		

Table employee_attendance working hour will be calculated depending on employees markin and markout timing at the end of the day.for public holiday the attendance status will be public holiday and full day working hours will be added for this day.calender date would be each day of the year for each employee to track the attendance status for each day whether he worked full day / half day / was on leave or it was a public holiday. and his payroll will be calculated salary per hr - every month based on working hours			
*Pk	id	integer	
Idx	employee_id	bigserial	

Table employee_attendance working hour will be calculated depending on employees markin and markout timing at the end of the day.for public holiday the attendance status will be public holiday and full day working hours will be added for this day.calender date would be each day of the year for each employee to track the attendance status for each day whether he worked full day / half day / was on leave or it was a public holiday. and his payroll will be calculated salary per hr - every month based on working hours			
Idx	mark_in_time	timestamp	
Idx	mark_out_time	timestamp	
	employee_type_id	bigserial	
Idx	attendance_status	varchar(1)	
Idx	working_hrs	integer	
	leave_type_id	bigserial	
Idx	calender_date	date	
Indexes			
Pk	pk_employee_attendance_id	id	
	idx_employee_attendance_employee_id	employee_id	
	idx_employee_attendance_mark_in_time	mark_in_time	
	idx_employee_attendance_mark_out_time	mark_out_time	
	idx_employee_attendance_attendance_status	attendance_status	
	idx_employee_attendance_working_hrs	working_hrs	
	idx_employee_attendance_calender_date	calender_date	
Foreign Keys			
	fk_employee_attendance (employee_id) ref employee (employee_id)		
	fk_leave_type_id (leave_type_id) ref leave_type (id)		

Table employee_history			
*Unq	employee_id	bigserial	
Idx	first_name	varchar(50)	
Idx	last_name	varchar(50)	
	address	varchar(150)	
	city	varchar(25)	
	country	char(25)	
	postal_code	integer	
	mobile_number	integer	
	alternate_contact_number	integer	
	educational_background	varchar(25)	
	designation_id	integer	
	work_experience	bigserial	
	experience_in	text	
	employee_dept_id	bigserial	
	branch_id	bigserial	
	nationality	varchar(25)	
	desk_id	varchar	

Table employee_history			
	payroll_id	bigserial	
	no_of_paid_leaves	bigserial	
	employee_manager_id	bigserial	
	employee_status	bool	
*Pk	id	bigserial	
	payslip	varchar(100)	
	employee_hr_id	bigserial	
Indexes			
Pk	pk_employee_history_id	id	
Unq	unq_employee_history_employee_id	employee_id	
	idx_employee_history_first_name	first_name	
	idx_employee_history_last_name	last_name	

Table employee_reimbursement			
*Pk	id	bigserial	
Idx	employee_id	bigserial	
	reimbursement_amount	float8	
	reimbursement_description	varchar(50)	
	reimbursement_date	date	
	category_id	bigserial	
	approval_status	char(1)	
Indexes			
Pk	pk_employee_bonus_id	id	
	idx_employee_bonus_reimbursement_employee_id	employee_id	
Foreign Keys			
	fk_employee_id (employee_id) ref employee (employee_id)		
	fk_category_id (category_id) ref category (id)		

Table employee_salary_history			
*Pk	id	bigserial	
Idx	payslip	varchar(250)	
	increment_history	float8	
	grade	varchar(5)	
	increment_year	date	
Idx	employee_id	bigserial	
Indexes			
Pk	pk_employee_payroll_hostory_id	id	
	idx_employee_salary_history_payslip	payslip	
	idx_employee_salary_history_employee_id	employee_id	
Foreign Keys			
	fk_employee_id_history (employee_id) ref employee_history (employee_id)		

Table employee_type employee types would be : internal employees,consultant employees,internship/thesis employees,External employees etc.			
*Pk	emp_type_id	bigserial	
	employee_type_name	varchar(100)	
	emp_type_description	text	

Indexes

Pk	pk_employee_type_emp_type_id	emp_type_id	

Table employee_type_mapping To identify the relationship of the external employees with their respective companies.			
	emp_type_id	bigserial	
	description	varchar(50)	
*Pk	id	bigserial	
	consulting_comapny_id	bigserial	
	contract_duration	varchar(50)	
	designation_id	bigserial	
	employee_id	bigint	

Indexes

Pk	pk_employee_type_id	id	

Foreign Keys

	fk_employee_type_designation (designation_id) ref designation (designation_id)	
	fk_employee_type (consulting_comapny_id) ref external_company_contract (contract_with)	
	fk_employee_type_employee (employee_id) ref employee (employee_id)	
	fk_employee_type_mapping (emp_type_id) ref employee_type (emp_type_id)	

Table external_company_contract any contract with outside company for the purpose of software purchase agreement or man power will be recorded in this table.			
*Pk	id	bigserial	
	company_name	varchar(50)	
	company_location	varchar(50)	
	contract_duration	float8	
	number_of_consultants	integer	
	contract_amount	float8	
Unq	contract_with	bigserial	

Indexes

Pk	pk_external_company_contract_id	id	
Unq	unq_external_company_contract_contract_with	contract_with	

Foreign Keys

	fk_external_company_contract (contract_with) ref company (branch_id)	

Table external_company_contract_history			
*	id	bigserial	
	company_name	varchar(50)	
	company_location	varchar(50)	

Table external_company_contract_history			
	contract_duration	float8	
	number_of_consultants	integer	
	contract_amount	float8	
	contract_with	bigserial	
*Pk	history_id	bigserial	
Indexes			
Pk	pk_external_company_contract_history_history_id	history_id	

Table job_openings			
*Pk	job_id	bigserial	
	job_position	varchar	
	job_description	text	
	is_active	bool	
	no_of_employee_required	integer	
	required_work_experience	text	
	skill_set	text	
Indexes			
Pk	pk_job_openings_job_id	job_id	
	idx_job_openings_job_id	job_id	

Table leave_request could be a leave or a comp-off request.comp-off : considering an employee works on a holiday then he/she can apply for comp-off leave which will go to manager for approval and on approval by manager, his paid leaves will be credited.			
*Pk	id	bigserial	
	employee_id	bigserial	
	from_date	date	
	to_date	date	
	leave_type_id	bigserial	
	comments	text	
	leave_approval_status	bool	
Indexes			
Pk	pk_leave_request_id	id	
Foreign Keys			
	fk_employee_id (employee_id) ref employee (employee_id)		
	fk_leave_type_id (leave_type_id) ref leave_type (id)		

Table leave_type leave types would be :for eg. paid leaves, sick leaves, comp-off.			
*Pk	id	bigserial	
	type_of_leave	varchar(25)	
	leave_description	varchar(50)	
	no_of_leaves	integer	
Indexes			
Pk	pk_leave_type_id	id	

Table leave_type leave types would be :for eg. paid leaves, sick leaves, comp-off.			
	idx_leave_type_id	id	

Table payroll			
*Pk	payroll_id	bigserial	
	dept_id	bigserial	
	salary_per_hr	float8	
Indexes			
Pk	pk_payroll_id	payroll_id	
	idx_payroll_payroll_id	payroll_id	
Foreign Keys			
	fk_payroll_department (dept_id) ref department (dept_id)		
	fk_payroll_company (payroll_id) ref company (company_id)		

Table public_holiday			
*Pk	id	bigserial	
	date_of_holiday	date	
	occasion	varchar(25)	
	company_state	varchar	
Indexes			
Pk	pk_public_holiday_id	id	

Table salary_increment it will be a master table where data would be like :Grade A -- 30%,Grade B -- 20%,Grade C -- 5%and indivisual employee will get a grade in employee tatble and his yearly increment will be done as per his grade and % increment decided for that grade by organziational management.			
*Pk	id	bigserial	
	grade	varchar(5)	
	%increment	float8	
Indexes			
Pk	pk_person_salary_increment_id	id	

Table ticket approval by is an optional field.For E.g :if employee needs any paid software to use for project purpose then first request needs to be approved by his manager.			
*Pk	id	bigserial	
Idx	assigned_to_employee_id	bigserial	
Idx	raised_by_employee_id	bigserial	
	ticket_description	bigserial	
Idx	ticket_status	bool	
	ticket_tat	date	
	ticket_approval_status	bool	
	approval_by	varchar(100)	
Indexes			
Pk	pk_ticket_id	id	
	idx_ticket_id	id	
	idx_ticket_ticket_status	ticket_status	

Table ticket approval by is an optional field.For E.g :if employee needs any paid software to use for project purpose then first request needs to be approved by his manager.			
	idx_ticket_assigned_to_employee_id	assigned_to_employee_id	
	idx_ticket_raised_by_employee_id	raised_by_employee_id	
Foreign Keys			
	fk_assigned_to_employee_id (assigned_to_employee_id) ref employee (employee_id)		
	fk_raised_by_employee_id (raised_by_employee_id) ref employee (employee_id)		